



André Ramos Jorge Correia

Licenciado em Engenharia Informática

Dados, Informação, Conhecimento, o Business Intelligence e as suas motivações

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática

Orientador: Paulo Orlando Reis Afonso Lopes,
Prof. Doutor, FCT/UNL-DI

Júri:

Presidente: Prof. Doutor Joaquim Francisco Ferreira da Silva, FCT-UNL-DI
Vogais: Prof. Doutor João Miguel da Costa Magalhães, FCT-UNL-DI
Prof. Doutor Paulo Orlando Reis Afonso Lopes, FCT-UNL-DI



FACULDADE DE
CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE NOVA DE LISBOA

Setembro de 2011



Título da dissertação

Dados, Informação, Conhecimento, o Business Intelligence e as suas motivações

Copyright

André Ramos Jorge Correia

A Faculdade de Ciências e Tecnologia e a Universidade Nova de Lisboa têm o direito, perpétuo e sem limites geográficos, de arquivar e publicar esta dissertação através de exemplares impressos reproduzidos em papel ou de forma digital, ou por qualquer meio conhecido ou venha a ser inventado, e de a divulgar através de repositórios científicos e de admitir a sua cópia e distribuição com objectivos educacionais ou de investigação, não comerciais, desde que seja dado crédito ao autor e editor.



Agradecimentos

Pretendo através destas linhas expressar a minha gratidão não só pela ajuda no desenvolvimento desta dissertação, mas também, pela carreira que tenho vindo a construir, a qual considero positiva.

Em primeiro lugar quero agradecer ao Professor Paulo Lopes, por me ter orientado no desenvolvimento desta dissertação, com a boa ajuda que me deu, quer através de correcções assim como pelas boas ideias que me foi transmitindo, e que, me permitiram atingir um resultado final mais rico.

Agradeço ainda aos colegas e amigos que conheci ao longo destes anos de carreira, e que, me permitem hoje, com satisfação, disponibilizar este trabalho. Posso agradecer em especial à Novabase, onde agradeço especialmente ao Fernando Jesus e ao João Martins por terem apostado em mim e pela minha evolução, ao Armando Mendes. Na Staples, à Carla Marques, ao Pedro Pimenta, Carlos Valente e João Silva. E na Maven onde se torna mais difícil agradecer a alguém em especial, pois toda a estrutura funciona como uma família, em que a entreajuda e a amizade é base para todo o resto, ainda assim, deixo um agradecimento especial ao Nuno Costa pela sua aposta na minha pessoa.

Por fim à Neuza por sempre me ter acompanhado e ajudado ao longo destes anos de profissão.





Nº do aluno: 36085

Nome: André Ramos Jorge Correia

Título da dissertação:

Dados, Informação, Conhecimento, o Business Intelligence e as suas motivações

Palavras-Chave:

- Inteligência
- Data Warehouse
- Integração
- Relatório
- Profissional
- Painel

Keywords:

- Business Intelligence
- Data Warehouse
- Integration
- Reporting
- Professional
- Dashboard





Resumo

O objectivo deste relatório é dar a conhecer, um possível percurso de carreira para um aluno que na entrada no mundo profissional, se interessou por sistemas de suporte à decisão. Descrevi a minha experiência profissional desde a entrada no curso de Engenharia Informática e as opções que fui tomando durante e depois do curso, que demonstram o interesse e tendência para esta área dentro das TI.

Assim, o facto de após terminar o curso, ter trabalhado em diversas áreas nas três empresas que integrei, desde o desenvolvimento aplicacional em ERP e CRM, na definição de requisitos de arquitecturas de HW e SW, na qualidade e integração de dados em múltiplos sistemas e em modelos analíticos, sempre em metodologias e ambientes diversos. Esta multiplicidade de cenários faz-me pensar que me encontro actualmente na área certa, o Business Intelligence.

O curso na FCT-UNL transmitiu-me entre outras, uma capacidade de adaptação ao nível das mais diversas tecnologias e metodologias, assim como uma grande tolerância à frustração, devido aos inúmeros obstáculos que foram sendo colocados durante o curso. Tudo isto tem permitido ao longo da minha carreira ultrapassar as sempre difíceis tarefas a nível técnico, funcional e de gestão que me foram surgindo.

Dados, Informação, Conhecimento e Inteligência é um fluxo natural e uma motivação para criar soluções de Business Intelligence. O fluxo descrito deve ser seguido em qualquer SI/TIC. Não é profícuo para nenhuma organização ter dados provenientes de inúmeras aplicações, se estes não gerarem informação que se traduza em conhecimento. O fluxo deve ser cumprido, para que o produto final de um projecto de Business Intelligence seja um acréscimo de conhecimento e valor para a organização ao qual pertence.

Espero transmitir claramente como funciona o mundo dos projectos de Business Intelligence, as suas componentes, dificuldades e particularidades.





Abstract

The purpose of this report is to demonstrate a possible career path for a student entering in the professional world and that is interested in decision support systems. I described my professional experience, since entering the course of Computer Engineering and the options that I took, either during the course or after. This shows my interest and trend for this area of IT.

So I think that after finishing the course, having the opportunity to work in various areas in the three companies which I was integrated in. I've done a lot, from application development to ERP and CRM, requirements for hardware and software architectures definition, data quality, data integration across multiple systems, analytical models and with different methodologies in various environments. That variety of scenarios made me believe that I'm currently in my vocational area, the Business Intelligence.

The course at FCT-UNL gave me the ability to easily adapt to diverse technologies and methodologies, as well as, a great resistance to frustration, due to the numerous obstacles that have been placed for us during the course. All this allowed that during my career, to always overcome difficult obstacles in technical, functional and management areas that still arises.

The theme chosen, data, information, knowledge and its connection to building business intelligence solutions, turns out to be a natural flow and should be followed by any IS/TIC. It is not useful, for any organization, to have data from many applications, if they don't generate information that can be translated into knowledge. This flow must be fulfilled so that the end product of a Business Intelligence project turns out to be an increase of knowledge and value to the organization which it belongs.

I hope to clearly show how does work the world of Business Intelligence projects, its various components, and specific difficulties.





Índice

Resumo	7
Abstract	9
Índice	11
Índice de Imagens	13
Glossário	15
1. Actividade Profissional	17
1.1 Novabase	20
1.2 Staples	20
1.3 BIMaven	21
2. Projectos mais relevantes.....	23
2.1 Projecto de criação de interface operacional na ACSS	23
2.1.1 Contexto	23
2.1.2 Requisitos e Motivação	24
2.1.3 Solução Técnica e Execução do Projecto	25
2.1.3.1 Planeamento	25
2.1.3.2 Arquitectura Macro	26
2.1.3.3 Infra-estrutura	27
2.1.3.4 Modelo Físico de estruturas de dados.....	27
2.1.3.5 Processos.....	29
2.1.3.6 Framework de carregamento	30
2.1.3.7 Processos de integração.....	35
2.1.4 Análise de Resultados	35
2.2 Projecto de criação de modelo analítico na ACSS	36
2.2.1 Contexto	36
2.2.2 Requisitos e Motivação	36
2.2.3 Solução Técnica e Execução do Projecto	36



2.2.3.1	Infra-estrutura	38
2.2.3.2	Arquitetura Macro e modelo físico	40
2.2.4	Análise de Resultados	51
3.	Análise aprofundada do percurso profissional.....	53
3.1	Iniciação - NB	53
3.1.1	PT-Sistemas de Informação.....	53
3.1.2	ESI – Espírito Santo Innovation	54
3.1.3	ESCOB – Espírito Santo Cobranças	56
3.2	Evolução - Staples	56
3.3	Consolidação - BIMaven	58
3.3.1	BIMaven.....	59
3.3.2	Media Capital	60
3.3.3	ACSS.....	61
3.3.4	Groupama Seguros.....	63
3.3.5	FNAC.....	63
3.3.1	EFACEC	64
3.3.1	ZON	64
4.	Referências.....	65
5.	Bibliografia adicional.....	67
	Anexos.....	69
1.	ODI	69
2.	OBIEE	72
2.1	Principais funcionalidades	72
2.2	Modos de visualização	74
2.3	Criação de Indicadores.....	76



Índice de Imagens

Figura 1 - Áreas Centrais de Actividade nas TI	17
Figura 2 - Cadeiras efectuadas durante a licenciatura da secção de SSDI	18
Figura 3 - Fases de carreira	19
Figura 4 – Entidades empregadoras durante a carreira	20
Figura 5 - Clientes por Empresa Empregadora	21
Figura 6 - Modelo de Alto Nível de Integração Operacional.....	23
Figura 7 - Quadrante Mágico de integração do Gartner Group para 2009	26
Figura 8 - Visão Macro do Projecto de Integração	27
Figura 9 – Infra-estrutura da solução.....	27
Figura 10 - Framework de carregamento a partir de RDBMS.....	33
Figura 11 - Framework de carregamento a partir de ficheiros	34
Figura 12 – Infra-estrutura da Solução do Modelo Analítico	39
Figura 13 – Arquitectura Macro do Modelo Analítico	40
Figura 14 - Quadrante Mágico de BI do Gartner Group para 2009 e 2010	43
Figura 15 - Framework Carregamento de Dimensões	48
Figura 16 - Framework Carregamento de Factuais.....	49
Figura 17 - Hierarquia de Tempo	51
Figura 18 - Arquitectura ETL vs EL-T	69
Figura 19 - Design Declarativo	70
Figura 20 - ODI Operations.....	71
Figura 21 - ODI Hot Pluggable	71
Figura 22 - Gráfico de Área.....	74



Figura 23 - Gráfico de barras horizontais.....	74
Figura 24 - Gráfico de Barras Verticais	74
Figura 25 - Gráfico Linear.....	74
Figura 26 - Gráfico de Pareto.....	74
Figura 27 - Gráfico de Passo.....	74
Figura 28 - Gráfico de Bolhas.....	75
Figura 29 - Gráfico de Radar	75
Figura 30 - Gráfico Circular.....	75
Figura 31 - Gráfico Gauge	75
Figura 32 - Gráfico Ticker.....	75
Figura 33 - Map Locations.....	75
Figura 34 - Map Prompt	76



Glossário

ACSS	Administração Central dos Sistemas de Saúde
AIX	Advanced Interactive eXecutive
AWK	Linguagem vocacionada para processamento de ficheiros de texto
BD	Base de Dados
BI	Business Intelligence
BMC	Empresa especializada em BSM
BSM	Business Service Management
CCB	Centro Cultural de Belém
CMMI	Capability Maturity Model Integration
CRM	Customer Relationship Management
CSV	Comma-separated values
DB	Database
DBMS	Database Management System
DM	Data Mart
DQ	Data Quality
DSV	Ambiente de Desenvolvimento
DW	Data Warehouse
EBCDIC	Extended Binary Coded Decimal Interchange Code
ERP	Enterprise Resource Planning
ETL	Extract, Transform & Load
FCT-UNL	Faculdade de Ciências e Tecnologia da Universidade Nova de Lisboa
FGAC	Fine-grained access control
FTE	Full-Time Equivalent
FTP	File Transfer Protocol
HP-UX	Hewlett-Packard Unix
HW	Hardware
ID	Identificador
IMAP	Internet Message Access Protocol
iSCSI	Internet Small Computer System Interface
ITIL	Information Technology Infrastructure Library
J2RE	Java 2 Runtime Environment
JDBC	Java Database Connectivity
JVM	Java Virtual Machine
KPI	Key Performance Indicator
MB	Megabyte
MDN	Ministério da Defesa Nacional
MS	Microsoft
MS IS	Microsoft Integration Services
MSTR	Microstrategy
NB	Novabase



OBIEE	Oracle Business Intelligence Enterprise Edition
ODBC	Open Data Base Connectivity
ODI	Oracle Data Integrator
ODS	Operational Data Storage
OLAP	Online Analytical Processing
OLTP	On-Line Transaction Processing
OS400	SO IBM desenvolvido para o AS/400
OTS	Oracle Transactional Storage
OWB	Oracle Warehouse Builder
PL/SQL	Procedural Language/Structured Query Language
Plain Text	Ficheiro com texto sequencial de fácil leitura
PM	Project Manager
PME	Pequenas e Médias Empresas
POP3	Post Office Protocol
PRD	Ambiente de Produção
PT	Portugal Telecom S.A.
PT-COM	PT Comunicações
RAC	Real Application Clusters
RDBMS	Relational Database Management System
RFD	Request For Development
RFI	Request For Information
ROI	Return Of Investment
ROLAP	Relational Online Analytical Processing
RTF	Rich Text Format
SAP	System Analysis and Program Development
SAP BW	SAP Business Warehouse
SCM	Supply-Chain Management
SI/TIC	Sistema de Informação/Tecnologias de Informação e Comunicação
SIG	Sistema de Informação de Gestão
SK	Surrogate Key
SLA	Service Level Agreement
SMS	Short Message Service
SNS	Serviço Nacional de Saúde
SO	Sistema Operativo
SOA	Service-Oriented Architecture
SPV	Serviço Pós-Venda
SQL	Structured Query Language
SSDI	Sistemas Simbólicos de Decisão e Informação
STA	Staging Area
SW	Software
TCO	Total Cost of Ownership
TELCOS	Telecomunicações
TI	Tecnologias de Informação
TST	Ambiente de Testes ou Pré Produtivo
VB	Visual Basic
VM	Virtual Machine
WPMS	Warehouse Physical Management System
XML	eXtensible Markup Language
ZPL	Z-level Programming Language

1. Actividade Profissional

Iniciei o meu percurso no mundo laboral e na área da Engenharia Informática ainda durante a faculdade, tendo na altura tentado aproveitar todos os conhecimentos que já trazia e os que fui adquirindo numa perspectiva de utilizador. Ainda assim, este início não teve qualquer relação com aquela que acabou por vir a ser a minha área central de actividade dentro das TI, o Business Intelligence.

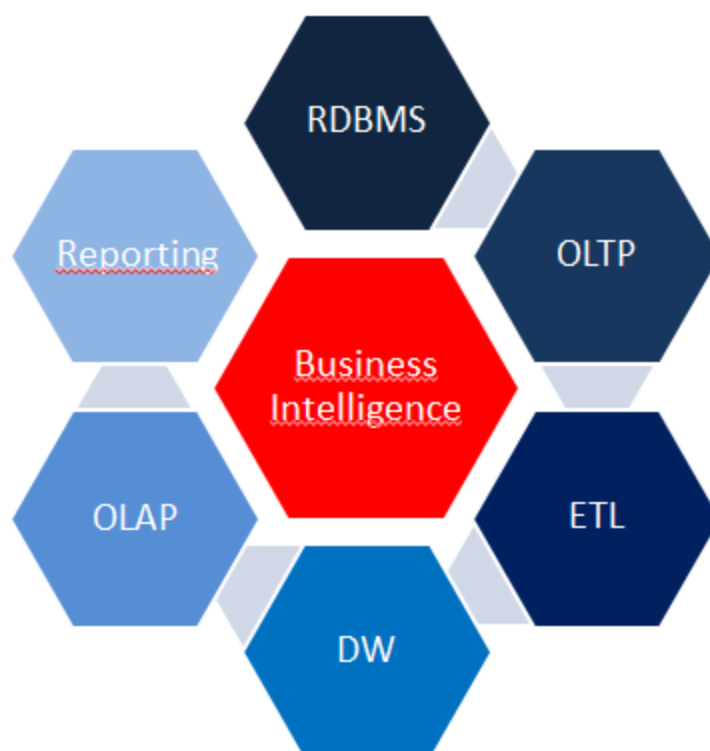


Figura 1 - Áreas Centrais de Actividade nas TI

A figura 1 representa de forma integrada as minhas áreas de actuação actuais e que circulam em volta da área de Business Intelligence. Assim dentro da sequência apresentada tenho:

- As RDBMS, utilizada na maioria dos projectos, entre elas, Oracle [1], MS SQL Server [2], Teradata [3], Sybase [4], DB2 [5] AS400¹[6], etc...
- A OLTP, parte integrante da maioria dos projectos como fonte de extracção de dados ou em alguns como destino.

¹ Minicomputador IBM com arquitectura EBCDIC, actualmente substituído pelos iSeries



- Os processos de Integração que denomino como ETL, e que, estiveram presentes em quase a totalidade dos projectos. Embora desde há alguns anos exista sempre uma tendência para os designar com esta sigla, cada vez fará menos sentido fazê-lo, pois começam a existir várias abordagens díspares e que não obedecem à mesma ordem nas operações.
- Os DW, que são objecto de todo o trabalho de modelação de base são grandes desafios. Estes estão sempre presentes nas grandes organizações e começam a fazer parte de algumas PME.
- Os modelos OLAP com todas as variantes existentes e que foram parte integrante de muitos projectos, tais como com o Microstrategy, o OBIEE e o Discoverer.
- Toda a camada de *reporting* e *dashboarding* assente no modelo OLAP pré-definido utilizando ferramentas como Microstrategy, o OBIEE e o BI Publisher.

Considero importante mencionar que de alguma forma existiu uma influência clara nas opções que efectuei durante a licenciatura na escolha do meu percurso. Relativamente às disciplinas opcionais que então elegi, demonstrei apetência para a secção central que refiro na figura 2, a SSDI.

A figura 2 dá a conhecer as cadeiras que efectuei da secção de SSDI [8]. Nos círculos vermelhos estão referidas as que eram obrigatórias no plano curricular, nos círculos verdes as que escolhi como opcionais. O projecto final está num círculo de cor amarela, pois embora seja obrigatório dentro do plano curricular, o seu conteúdo acaba por ser opcional, e também aqui, a minha opção acabou por recair naquele que foi o meu primeiro projecto de Business Intelligence.

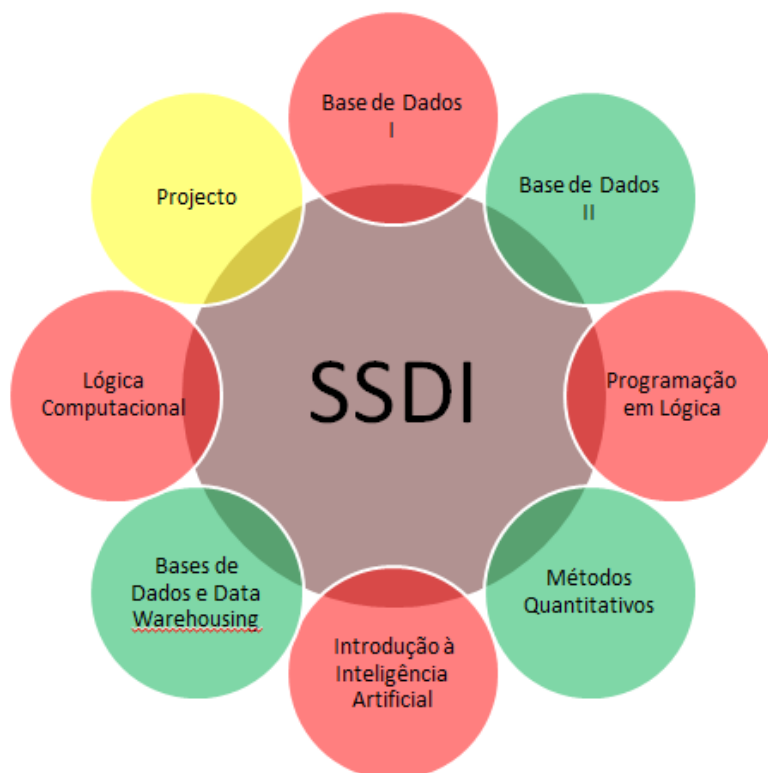


Figura 2 - Cadeiras efectuadas durante a licenciatura da secção de SSDI

De seguida apresento o resumo do meu percurso dentro da Engenharia Informática.

Iniciei a minha actividade, ainda durante o curso, no apoio a várias empresas e particulares presentes em vários sectores de actividades, realizando as seguintes tarefas, entre outras:

- Montagem de máquinas e *storage*;
- Instalação e configuração de sistemas operativos MS-DOS, Microsoft Windows e Linux;
- Instalação e configuração de inúmero *software* necessário na área de actividade de cada cliente desde SW de restauração, contabilidade e arquitectura;



- Desenho de redes e instalação das mesmas, desde a passagem de cablagem entre divisões; acoplagem de fichas *Ethernet* RJ45. Instalação de *Switch* e *Routers* e sua posterior configuração;
- Gestão de sistemas distribuídos de *storage* e impressão em redes Microsoft [9] e Novel [10] ;
- Reparação de máquinas.

Este processo anterior ao término do curso de Engenharia Informática deu-me uma noção muito importante do funcionamento das máquinas e reconhecimento dos problemas de instabilidade no seu comportamento, do cuidado a ter com os equipamentos com os quais é efectivamente necessário ser providente e assim evitar problemas directamente associados ao HW. Para além disso conheci vários SO e também a forma de interagir com o seu *core* o que, me ajuda ainda hoje na forma como abordo um sistema que me é desconhecido.

Durante o ano de 2004, quando estava na fase final do curso na FCT-UNL procurei fazer o trabalho final numa empresa da área das TI, particularmente na Consultoria. Isto deveu-se ao facto do meu objectivo ter sido sempre trabalhar no meio empresarial e principalmente ao ser consultor, poder ter acesso a inúmeros tipos de actividades e negócios e aprender um pouco sobre cada um deles. Assim, quando surgiu como hipótese na lista de estágios propostos entrar na NB, uma empresa de renome não só a nível nacional mas também internacional, resolvi propor-me para o mesmo.

Após uma entrevista e analisadas as candidaturas, fui aceite e iniciei o meu estágio numa empresa do grupo que apostava especificamente numa área em que eu sempre procurei estar colocado, e que, ainda hoje me continua a fascinar, pelo que é possível produzir, o BI.

Aquilo que sempre me interessou no BI foi a capacidade de pegarmos num amontoado de sistemas que habitualmente existem nas organizações, seleccionarmos e consolidarmos a informação que realmente é importante para coordenar os objectivos e decisões estratégicas das mesmas. No fundo, acabamos por ter um papel fundamental nas intervenções que levamos a cabo, seja em alturas de prosperidade económica aquando os mercados são mais exigentes e esperam índices elevados de rendimento e rentabilidade na sua produção ou ainda nas alturas de crises financeiras que por questões de sobrevivência é muito importante conhecer a realidade da organização que gerimos.

Assim, este primeiro projecto - o “Basileia II”², foi fundamental para perceber como poderia intervir concretamente a nível do negócio de uma grande organização. Esta versão do acordo cuja entrada em vigor aconteceu em 2007 permitiu às entidades financeiras um conhecimento mais profundo dos seus clientes. Foi esta vantagem competitiva que passou a permitir-lhes medir o grau de risco associado a cada cliente. Esta informação é uma mais-valia nas suas operações diárias, sendo que para a poder atingir será necessário possuir um repositório contendo o histórico das suas operações, além de informação adicional proveniente de fontes externas, de modo a posteriormente estimar os diversos riscos presentes em cada operação.

Após cumprir o projecto de final de curso, fui convidado para integrar os quadros da NB. Daqui em diante consigo dividir a minha carreira em 3 fases, que coincidiram com os momentos de transição para novas entidades empregadoras, dando origem a um novo conjunto de desafios e responsabilidades.

Assim o meu percurso terá seguido as seguintes fases:



Figura 3 - Fases de carreira

² O **Acordo de Capital de Basileia II**, foi um acordo assinado no âmbito do Comité de Basileia em 2004 para substituir o acordo de Basileia I. Este consiste em 3 pilares (Capital, Supervisão, Transparência e Disciplina de Mercado) e 25 princípios básicos sobre contabilidade e supervisão bancária.



Estas fases terão coincidido em termos temporais:



Figura 4 – Entidades empregadoras durante a carreira

1.1 Novabase

Como consultor na Novabase [11] Business Intelligence assumi desde o início o papel de *Analist & Developer*. Isto permitiu-me, devido à complexidade e versatilidade exigidas, desenvolver fortemente a componente técnica nas áreas de Modelação de dados para DW, desenvolvimento de processos ETL e conhecimentos específicos de componentes de *reporting*.

A experiência na consultoria permite uma facilidade na compreensão dos problemas e requisitos apresentados pelos diversos clientes, assim como a necessidade destes obterem soluções de qualidade, sempre com pensamento na evolução dos sistemas.

1.2 Staples

Após a saída da Novabase seguiu-se uma experiência numa vertical na área do Retalho.

Na Staples [12] trabalhei no seu *Head Office* embora por vezes fizesse projectos específicos para lojas que me solicitavam a deslocar-me às mesmas. Foram-me atribuídas novas funções tendo sido possível atingir um maior conhecimento da interacção de todos os sistemas e conhecer *on field* todo o fluxo de passagem de informação. Desde os POS nas lojas passando pelas interfaces de integração específicas nas RDBMS dos sistemas aplicativos mais básicos, a integração no ERP, CRM, e por fim nos modelos analíticos.

Na Staples acabei por ter uma interacção diferente da que conhecia com fornecedores de Sistemas, Produtos e Hardware. Estava no papel de comprador e não de vendedor e a minha visão mudou radicalmente na análise das necessidades dentro de uma organização. As minhas responsabilidades consistiam agora na definição e especificação de requisitos, disponibilização de ambientes de DSV e TST e efectuar testes e validar as soluções. Mais tarde aquando do meu regresso a Consultoria considero que esta terá sido umas das grandes mais-valias de ter trabalhado numa vertical, compradora de recursos externos. Pois quem só trabalhou como consultor, não tem noção de qual o papel desempenhado pelos elementos que mantêm os SI de uma organização. Hoje, e após ter assumido tais funções penso que atingi maturidade na relação com os SI dos clientes com os quais tenho de lidar.

Outra inovação foi o conhecimento no desenho de novas soluções *end-to-end*. Aprendi a combater e acompanhar as restrições naturais deste tipo de soluções, que surgem seja por questões de gestão a vários níveis, como sejam *budget*, espaço ou necessidades inerentes á própria estratégia da organização, a médio ou longo prazo.

Consegui alavancar os meus conhecimentos, e, assim ter uma noção mais próxima da realidade da relação entre o ROI e a gestão de necessidades em termos de *timeline*.

1.3 BIMaven

Com o regresso à Consultoria, integrei a BIMaven Consulting [13] a partir de Dezembro de 2008. Uma *startup* que tinha arrancado com a sua actividade em 2007 e cujo *core* era o BI. Em termos de nível de carreira assumi imediatamente as funções de *Senior Consultant*.

O meu contributo até a actualidade focou-se em várias tecnologias no mundo do Business Intelligence e Integração de dados, sendo que em termos de área de negócio embora tenha passado por várias áreas, em termos temporais centrou-se na Saúde. Um dos objectivos que me foi proposto foi arrancar e fazer crescer uma parceria com a Oracle na sua área de *Middleware*, uma vez que até então não estaria deliberadamente a ser potenciado o negócio ou projectos directamente com estes.

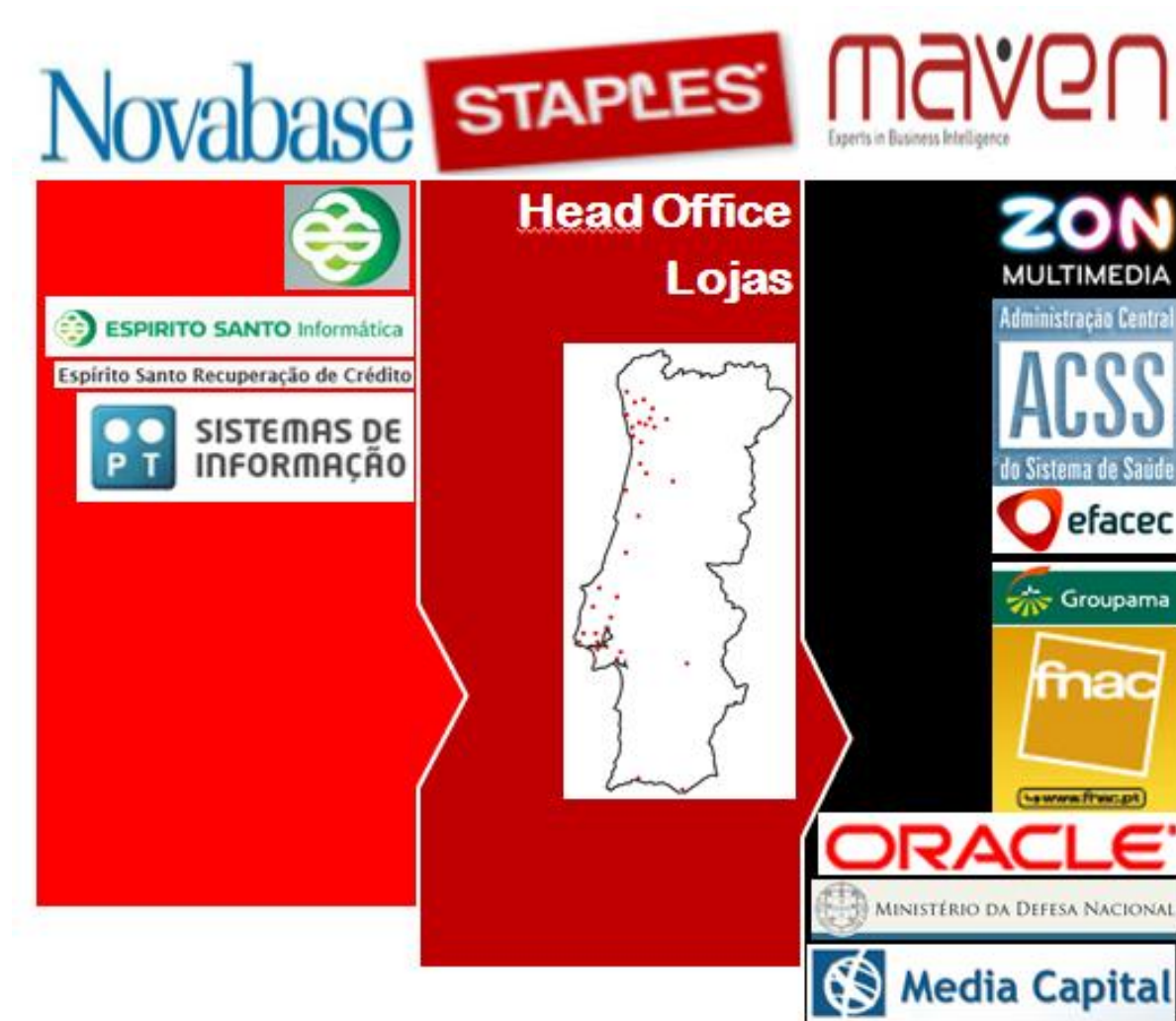


Figura 5 - Clientes por Empresa Empregadora



2. Projectos mais relevantes

2.1 Projecto de criação de interface operacional na ACSS

2.1.1 Contexto

Este projecto tinha como principal objectivo a criação de uma nova *interface* para um sistema transaccional. Esta *interface* funcionava entre aproximadamente 80 repositórios externos alimentados por uma aplicação criada em Oracle Forms, e um repositório central.

Assim, a tecnologia utilizada, as *Stored Procedures* em PL/SQL iria ser substituída por uma ferramenta visual de integração, o ODI, por forma a garantir o controlo e automatismo total no processo de carregamento e, consequentemente o mínimo de intervenção humana possível.

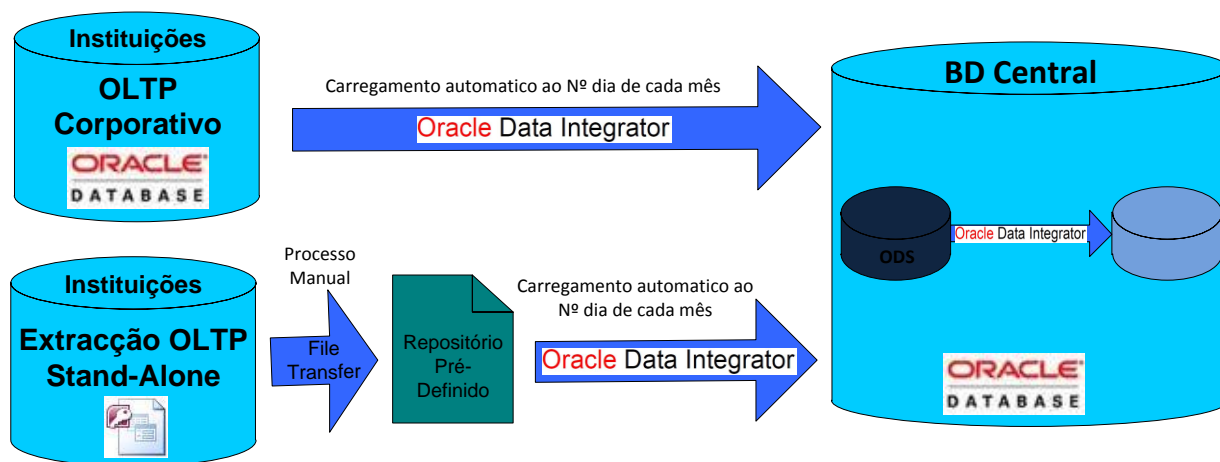


Figura 6 - Modelo de Alto Nível de Integração Operacional

Os automatismos criados deveriam permitir um aumento de produtividade, libertando assim as equipas de manutenção de tarefas rotineiras e consumidoras de tempo, proporcionando a possibilidade de se focarem noutras mais de acordo com as suas funções.

Esta é uma *interface* crítica para a organização, pois o repositório central, destino deste processo de carregamento, passou a ser fonte de um sistema analítico, o que obrigou a um maior controlo da qualidade e disponibilidade da informação a ser carregada devido a necessidade de maior consistência da informação, fiabilidade e integridade.

Em termos do que foi proposto, as características do novo processo foram:

- Funcionamento de forma totalmente automática inclusive na gestão de falhas;
- Qualquer intervenção humana, quando necessária, seria solicitada. Isto garantia a não alocação de pessoas ao controlo durante o tempo em que decorresse o processamento;



- Criação de uma *framework* de controlo de processos comum a todos os projectos gerados na nova ferramenta de integração. Esta consistia na criação de tabelas de Log dos processos na contagem de extracções e carregamentos;
- Gestão automática de registos rejeitados com a geração de mensagens de aviso;
- Controlo de falhas e erros ao longo de todo o processo;
- Utilizadores com *login* de acesso específico, aos quais só seriam permitidas determinadas tarefas na ferramenta de integração, tais como, controlarem a actividade na execução dos processos. Passou a ser possível manter um elevado nível de segurança no acesso à informação que iria ser processada, os *Audit-Trails*.

2.1.2 Requisitos e Motivação

O modelo operacional apresentava-se limitado, não tanto pela sua configuração, mas essencialmente pelas tecnologias envolvidas, neste caso o *scripting* PL/SQL Oracle com *Stored Procedures*.

A integração através de *scripting* é habitualmente olhada como uma solução simples, barata, rápida e ao alcance de todos. Mas esta visão está comprovada como sendo errada, quer através dos custos de manutenção a longo prazo ou do facto de não ser fácil reutilizar o código gerado. A complexidade dos processos de integração, que actualmente são sujeitos a SLAs relacionados com a disponibilidade e alta qualidade da informação, onde se exige uma *metadata* robusta, também não se coaduna com esta opção. O *scripting* além de extremamente inflexível resulta habitualmente em redundância de processos, recursos subaproveitados e custos elevados habitualmente ligados a infra-estruturas, desenvolvimento, suporte, manutenção e gestão.

Em termos de requisitos, neste projecto existiam os seguintes:

- Substituir o processo antigo onde eram utilizados scripts PL/SQL Oracle, ou seja, uma tecnologia utilizada de difícil manutenção. Esta obrigava a intervenções demoradas sempre que se revelava necessário efectuar alguma alteração ou correcção;
- A ferramenta de integração escolhida teria que poder aceder a BD Oracle 7.3, visto serem estes os repositórios operacionais que seriam fontes do processo;
- A ferramenta de integração escolhida teria que gerar alertas e avisos relativos ao processamento através do envio de *mails*. Para isso seria necessário a garantia de que o servidor onde a ferramenta de integração se encontrasse, tivesse acesso ao servidor de *mail*, para poder gerir o envio de *mails* despoletados pelos processamentos.
- Solucionar a integração da informação que não podia ser por via *DB-Link*. Devido ao facto de provir de diferentes aplicações que era recebida em ficheiro, convertida manualmente com Access e só posteriormente carregada.

Em termos de motivação, havia inúmeras vantagens nesta nova solução, tais como:

- Eliminar ao máximo a intervenção humana na totalidade da execução e monitorização dos processos, dispensando o processo de carregamento de ser efectuado por técnicos com um conhecimento profundo;
- Optimizar o processo de validação da informação carregada para ser efectuado apenas no final do carregamento. Esta questão criava grandes perdas de tempo devido ao facto de que sempre que fosse detectado algum erro, só o seria no final do processamento, o que por vezes obrigava a reprocessamentos totais ou parciais;



- Controlar e tratar os erros resultantes do processo, dado que este era rudimentar e pouco abrangente. Não existiam estruturas de *logging* durante a execução, apenas contagens de registos carregados no final do processamento;
- Incrementar a controlo nos acessos a BD ou nas operações sobre a mesma, eliminando vulnerabilidades relativas à informação existente no repositório central.

2.1.3 Solução Técnica e Execução do Projecto

2.1.3.1 Planeamento

Para iniciarmos o projecto criei um plano no *MS Project* onde referi todos os pré-requisitos necessários para o arranque do mesmo. Defini que iriam existir 3 ambientes de BD e 3 aplicativos, o de Desenvolvimento, o de Testes ou Pré-Produção e o Produtivo. Esta metodologia que havia seguido em outros projectos e clientes proporciona normalmente maior qualidade nos *deployments* em produção.

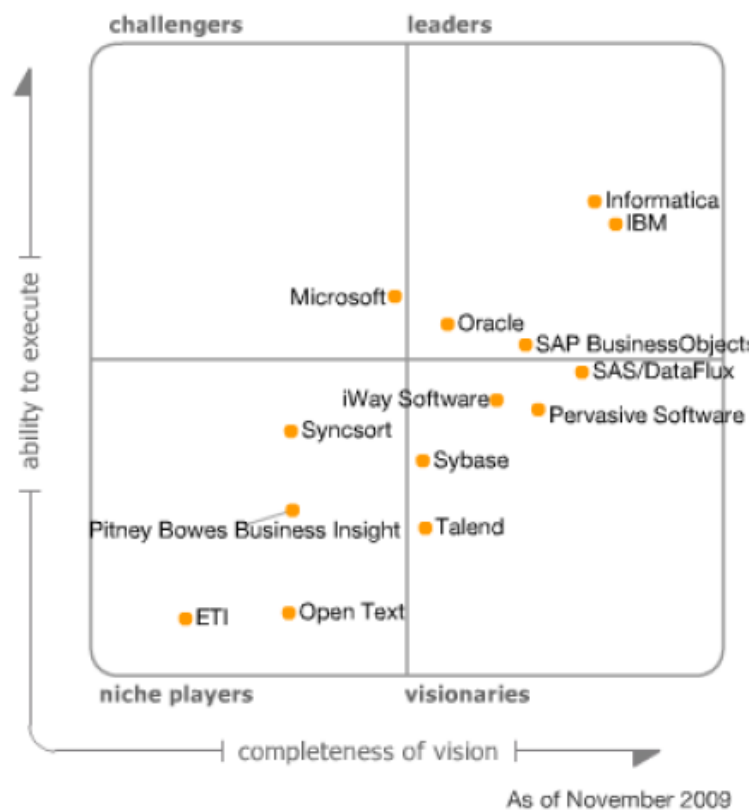
Os ambientes têm os seguintes objectivos:

- Ambiente de DSV, onde é feita a criação, exploração e testes unitários;
- Ambiente de TST, onde fazemos os chamados testes funcionais, performance e integração. Utilizam-se dados provenientes de PRD, como excepção, devendo-se por questões de segurança dos dados ou de confidencialidade da informação contida criar dados de teste mascarados e com as condições exigidas pelas novas especificidades.
- Ambiente de PRD, onde são efectuados os testes de aceitação, fazendo com que o produto passe a PRD com um risco mínimo e garantia de utilização imediata.

Tendo em conta que o projecto foi totalmente desenvolvido nas instalações do cliente, foi efetuada uma preparação a nível dos pré-requisitos para arranque, entre estes:

- Disponibilidade do servidor de integração instalado e configurado com acesso a todos os utilizadores de BD necessários. Era um processo que em termos de ambientes tinha uma janela de tempo que ia acompanhando o mesmo, ou seja no arranque do projecto apenas se exigia o ambiente de Desenvolvimento nas condições descritas;
- Carregamento de informação proveniente de uma das fontes operacionais nos três ambientes;
- Acesso a todas as fontes operacionais.

Antes de iniciar a descrição sobre os desenvolvimentos e a infra-estrutura vou explicar a razão pela qual a ferramenta de integração escolhida foi o ODI em detrimento de muitas outras disponíveis no mercado. A Oracle era um líder no “Quadrante Mágico da Gartner de Ferramentas de Integração”[14] em 2009. O que à cabeça podia ser suficiente por parte do cliente para fazer esta opção mas a principal razão foi a conectividade a BD Oracle 7.3, isto porque todas as fontes operacionais do processo tinham esta versão de BD, esta situação condicionava logo a cabeça a escolha embora claro existisse concorrência, como o Informatica Power Center, líder no quadrante referido. Ainda assim o ODI apresenta algumas características próprias que o distingue da restante oferta do mercado e que estão descritas, no anexo nomeado ODI, as que considero estarem directamente ligadas com o desenvolvimento deste projecto.



Source: Gartner (November 2009)

Figura 7 - Quadrante Mágico de integração do Gartner Group para 2009

A componente de *logging* e gestão do processo também teve que ser criada neste projecto com o objectivo de ser genérica e transversal a todos os projectos desenvolvidos em ODI. Só assim conseguiríamos tirar partido total do investimento na tecnologia, potenciando assim a sua utilização no futuro em novos projectos e inclusive em outra áreas. Esta abordagem é considerada uma mais-valia no panorama das *best practices* dos projectos de integração dentro de uma organização.

2.1.3.2 Arquitectura Macro

Assim, em termos de arquitectura macro, a solução é apresentada em seguida. Existem inúmeras fontes operacionais, na sua maioria RDBMS Oracle 7.3 e as restantes a partir de ficheiros extraídos com uma estrutura igual às tabelas provenientes das RDBMS.

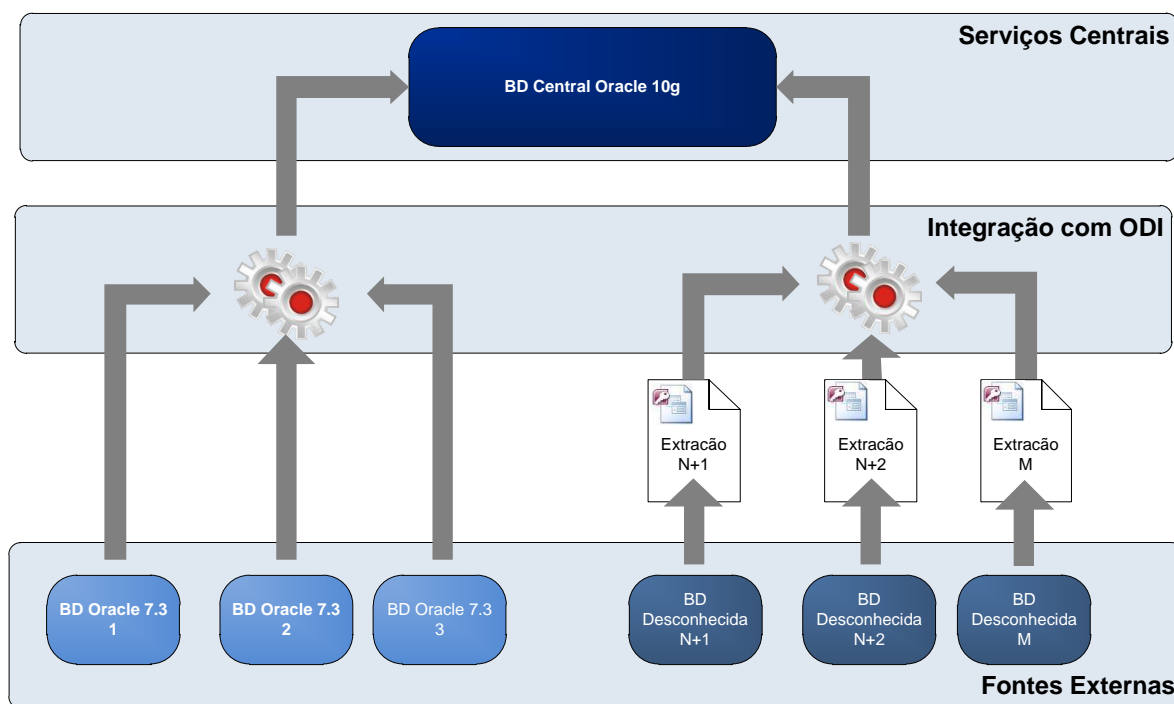


Figura 8 - Visão Macro do Projecto de Integração

2.1.3.3 Infra-estrutura

Para este projecto utilizámos três máquinas, a infra-estrutura em termos de Hardware era desconhecida por mim e por esta razão, apenas posso apresentá-la ao nível de Software:

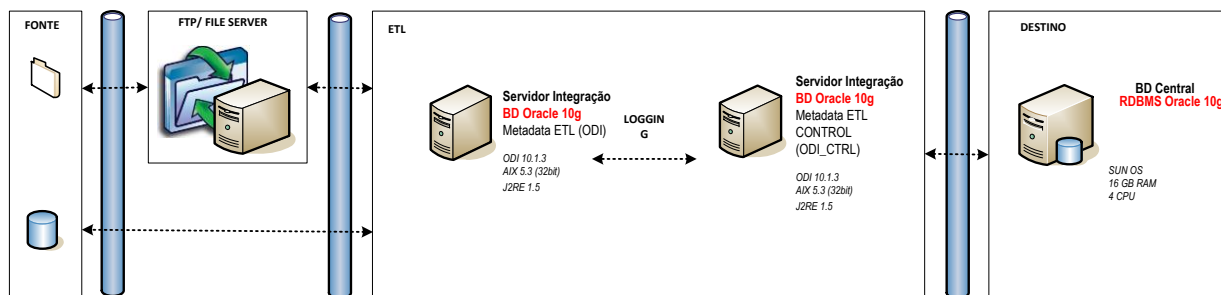


Figura 9 – Infra-estrutura da solução

2.1.3.4 Modelo Físico de estruturas de dados

Quanto ao modelo físico, neste projecto era constituído por 3 componentes perfeitamente distintos, a saber:

- RDBMS Oracle 7.3
Esta era a fonte operacional do processo de integração. Foi disponibilizado um utilizador de BD ao qual poderíamos ter acesso via JDBC utilizando o conector específico do ODI. A única necessidade aqui seria o pedido a cada uma das instituições onde se encontravam estas RDBMS, de abertura na sua rede da porta 1521 a qual o ODI necessita para poder aceder ao



agente que contem a configuração para o utilizador de BD destino. Este utilizador tinha uma estrutura pré-definida em termos de tabelas e não poderia ser alterado enquanto decorressem os desenvolvimentos no projecto.

A informação era disponibilizada nestas fontes até um dia pré-definido de cada mês e nesse dia a *interface* de carregamento era despoletada efectuando uma sequência de validações de forma a determinar se era possível fazer o carregamento e só após estas era então iniciada a extracção da informação.

- Estrutura de ficheiros CSV em *Plain Text*.

Eram disponibilizados mensalmente e através de *mail* um grupo de ficheiros com uma estrutura pré-definida que iam sendo colocados numa determinada data, num directório específico, na máquina onde se encontrava o servidor de integração. Existia um directório na respectiva árvore que era adjudicado ao projecto, sendo que neste foram criadas directorias específicas nomeadas com a sigla da organização que enviava os ficheiros. Este processo tinha aqui uma componente manual na qual era necessário associar o elemento humano para a cumprir todos os meses. Este teria que descarregar os ficheiros de um *mail* específico e posteriormente via FTP transferir os mesmos para o directório pré-definido. Esta foi a única tarefa manual que ficou associada ao projecto, devido não só ao facto de este ser um processo alternativo e com um grau de utilização baixo, relativo à integração por JDBC como também por motivos organizacionais. Foi proposto como alternativa a este processo a utilização da *task* do ODI nomeada ODIREADMAIL, que permitia através dos protocolos POP3 ou IMAP, a recepção e tratamento de *mails* e de todas as suas componentes

Neste caso iríamos utilizar o *subject* do *mail*, como identificador da instituição, para criarmos o directório onde seriam colocados os ficheiros que estavam em *attachmment*. O referido directório encontrava-se na mesma máquina física, ou seja no servidor de integração. Após efectuada esta tarefa as posteriores acções correspondem às descritas anteriormente.

- RDBMS Oracle 10G

Tirando proveito deste projecto, foi decidido fazer a migração de Oracle versão 8i.

Foi feito um processo de *export* da RDBMS 8i e posteriormente um *import* para a 10G, com o objectivo de disponibilizar a informação necessária para o desenvolvimento do projecto. Pelo facto de só existir um ambiente, este foi utilizado para desenvolvimento, testes e no final do projecto passou a produtivo. Esta migração de versões de BD constituiu também uma forma de termos um ambiente dedicado para o projecto.

Seguindo metodologias de desenvolvimento de projectos de integração nos quais participei anteriormente, foram criados nesta RDBMS dois utilizadores, um para o ODS, e outro, o destino final dos processos de integração. A vantagem da criação de um ODS reside no facto de um utilizador de BD estar dedicado a esta tarefa específica de recepção de informação. Numa RDBMS Oracle, a tarefa menos consumidora de tempo e recursos de máquina é a inserção de dados. Tendo em conta que num processo de integração com entidades externas existem limitações temporais para a concretização da operação de recepção de informação, precisamos de um processo o mais leve possível para a máquina em utilização. Desta forma, diminuámos o impacto nos canais de comunicação utilizados e nas regras de gestão da própria rede. Existem regras em organizações que não permitem a abertura de conexões externas e muito menos por tempos indeterminados com máquinas específicas. Assim as boas práticas sugerem que num ODS o acesso quando efectuado a uma determinada estrutura deve ser o mais célere possível. A operação de *insert*, em termos da segurança lógica da BD actua de um modo transparente para a ferramenta de integração. Esta vai gerar um *Lock* sobre a estrutura onde estiver a inserir, e durante o tempo em que decorre a operação. Quando é feito o *unlock*, as outras operações que estiverem na fila para serem efectuadas vão aceder a BD e assim, colocar o menor peso nos recursos da máquina.

Em termos das estruturas destino e pelo facto da volumetria de dados ser elevada optamos por uma solução de particionamento das tabelas a carregar. A decisão de particionar ou não uma tabela deve-se essencialmente a questões de *performance*. O conceito de particionamento numa tabela é um conceito lógico, o que significa que no utilizador de BD vamos ver apenas



uma tabela mas em termos físicos e na gestão da própria RDBMS vão existir tantas tabelas, quanto o número de partições e "subpartições" que sejam criadas para a mesma. Esta operação vai permitir alocar partições em *tablespaces* que inclusive poderão ser colocados em *storages* diferentes o que vai constituir numa grande ajuda na gestão deste e vai resultar num incremento de *performance*.

Na escolha das chaves de partição, optamos na sua maioria por chaves compostas contendo a instituição e o ano aos quais a informação respeitava. A razão prende-se com o facto de termos informação com cerca de 20 anos e referente a um total aproximado de 80 instituições. Relativamente à indexação, na gestão da BD Oracle e ao nível do particionamento, nas versões posteriores a 10G deixou de ser necessária a criação de índice para as chaves de partição, porque na criação desta está associada a operação de criação do índice. Assim sendo a própria BD vai gerir e actualizar este sempre que forem adicionadas ou removidas partições.

2.1.3.5 Processos

Este processo de integração tinha algumas particularidades, entre elas:

- Existência de registos de substituição nos carregamentos. Ou seja, existiam registos com uma chave que se encontrava integrada e com a entrada de um novo registo com a mesma chave iria provocar a sua substituição. Para este processo ser limpo na visão de quem recebe, de forma a manter a integridade da informação, esta era enviada uma estrutura com códigos que deviam ser eliminados do utilizador de BD Central.

Este processo, de forma a não impactar com o de carregamento era executado antes do mesmo. Assim todos os registos que deviam ser eliminados eram marcados neste passo, e aquando do processo de carregamento estes já teriam sido eliminados, não sendo gerados problemas de integridade no sistema.

- Num SI, existem utilizadores a quem é dada autorização para criar, ler, actualizar e apagar. Assim, foram criadas regras específicas do lado das fontes do processo, para que, fosse possível do lado de quem apenas queria ler a informação saber, se a poderia carregar. Isto significa que o delta [15], referente ao mês em questão, teria que estar fechado e a informação disponibilizada sem problemas.

As estruturas fonte e destino do ODS eram exactamente iguais, com a *nuance* de ser incluído do lado do ODS um único campo adicional onde era colocada a data de processamento, e que, iria servir de base no decorrer do mesmo. A validação efectuada no arranque do processo referia-se a uma estrutura de *logging* do lado da fonte Oracle 7.3, que havia sido criada para consultas por parte das interfaces de leitura. Nesta existia apenas um campo (*flag*) que referia se a informação estava correcta e estabilizada, se poderia ser carregada ou ainda se deveria abortar imediatamente o processo. Esta *flag* era validada no arranque do processo.

A situação descrita refere-se a uma problemática que experienciei em vários projectos semelhantes, e que está relacionada com a definição de responsabilidades aquando da criação de novas interfaces entre sistemas, principalmente quando os seus *owners* são distintos. A *best practices* neste sentido, e pelas quais sempre me regi é, que quem "chega" tem que se adaptar ao ambiente já existente, isto sempre dentro de um bom senso. Trata-se de uma questão que não tem a ver tanto com engenharia informática mas com um código deontológico. Por vezes por abusos de poder, estas questões são ultrapassadas sem qualquer rigor ou lógica e o resultado habitualmente é, instabilidade causada não só nos novos sistemas como ainda nas suas fontes.

Não significando que não possam existir adaptações por razões de necessidade de nova informação ou até de formatos na informação transferida mas esta situações devem habitualmente ser vistas como excepcionais, assim diminuámos ao máximo os riscos inerentes ao projecto para que decorra com sucesso. Por esta razão devem ser evitadas todas as variáveis nas quais o controlo não esteja totalmente do lado de quem está a realizar a nova implementação.



- Este processo é executado todos os meses e é totalmente modular ao mais baixo nível de granularidade. Desta forma é sempre possível, em caso de falha em qualquer ponto, reexecutar o processo a partir daí e não na totalidade. Estes pontos de “reexecução” foram escolhidos de acordo com a sequência global do processo e das dependências existentes. Isto significa que antes de executar um *package* vão ser verificadas as condições para o arranque. Esta validação é feita em concordância com os seus precedentes. Assim em caso de falha de qualquer um dos seus precedentes, o *package* não estará em condições de ser executado.

2.1.3.6 Framework de carregamento

Todos os processos de carregamentos tiveram como base uma *framework* específica, por onde se regeu todo o fluxo de cada processo. Tratou-se de uma *framework* que desenhei com base na experiência que fui acumulando ao longo de vários processos de integração em que participei e com alguma informação que fui recolhendo na Web sobre esta matéria.

A experiência acumulada neste tipo de processos de integração transaccional ocorreu principalmente na Staples onde lidei várias vezes com problemáticas nos processos de *night* e *day run*. Dependendo da forma como vai ser feita a integração, as *frameworks* definidas têm que ser adaptadas à forma como interagem com a informação e também, dependendo das RDBMS que estiverem em causa. Para este projecto criei duas versões, uma para carregamento utilizando como fonte as estruturas de uma RDBMS e outra para utilizar como fonte, ficheiros *Plain Text*.

Qualquer das *frameworks* criada inclui a *framework* de controlo de processos em comum. Esta é genérica para todos os processos gerados, e está disponível num repositório central onde se encontra toda a *metadata*³ do ODI. Através desta última, foi construído o controlo do fluxo dos vários processos e respectivas interdependências, e ainda a geração de mensagens de erro, que são escritas em tabelas de *logging* e posteriormente utilizadas na criação de *mails* informativos para os gestores do processo.

A Framework para carregamento a partir da RDBMS pode ser visualizada na figura 10 e consiste nos seguintes passos:

1. É feito um *delete* às tabelas da STA por Instituição a carregar;
2. Existem dois tipos de tabelas disponíveis, as que constituem o universo de informação a ser carregada, na forma de delta de carregamento, e as que contêm os episódios a serem eliminados de toda a estrutura operacional. O conceito de delta de carregamento é a nova *tranche* de dados adicionais a serem introduzidos na BD limitados inferior e superiormente por data;
3. A partir das RDBMS das fontes e num utilizador destinado apenas a leitura de interfaces é efectuado o carregamento directo das tabelas com a mesma nomenclatura relativamente a STA, onde vamos centralizar a informação. Este carregamento será efectuado através de operações de leitura e escrita directa sem efectuar qualquer tipo de transformação. Assim viabilizamos a máxima *performance* na leitura à BD das Instituições e o mínimo tempo possível para os integrar.
4. O processo subsequente tem como fonte a tabela com os registos a eliminar. Esta tabela contém as chaves a partir das quais podemos seleccionar os registos que serão eliminar. A partir das chaves definidas, é feito um *update* para marcação dos registos a eliminar;
5. No passo anterior os registos foram marcados, e são agora copiados para as respectivas tabelas de histórico. Tabelas que têm a mesma estrutura que as de destino do processo operacional, excepto na inclusão do campo correspondente à data de remoção, e na nomenclatura do seu nome onde foram prefixadas por “REM_”;
6. Neste passo vamos eliminar das tabelas finais os registos marcados para eliminar no passo 4 através de um processo de delete por chave de instituição e indicador de remoção;

³ Dados acerca de um ou vários aspectos dos próprios dados.



7. Agora são aplicadas todas as regras de transformação e integridade efectuando o carregamento a partir das tabelas do ODS para o seu *schema* final. Este processo teve em paralelo validações onde todos os registos que não estivessem em conformidade com as regras definidas serão inseridos com o respectivo código de erro nas tabelas com o mesma nomenclatura que as de origem mas prefixadas com “ERR_”. Com isto, após o *término* do processo e a partir das tabelas prefixadas com “ERR_” será fácil analisar problemas na integridade da informação ou do próprio processo;
8. Neste processo, é feita uma comparação entre a informação que está pronta para sair da STA e entrar nas tabelas finais. Nesta fase, a partir da data de processamento, em cada nova inserção será actualizado o campo de data de inserção e de alteração ou *update*;
9. Após o final do processo de carregamento e quando confirmado que decorreu bem, serão eliminados os registos existentes nas tabelas fonte ao processo.

A Framework para carregamento de ficheiros pode ser visualizada na figura 11 e consiste nos seguintes passos:

1. Após serem recebidos por *email* proveniente de cada instituição, o envio dos ficheiros é efectuado via FTP para uma pasta específica com o nome da instituição e a partir da qual foram enviados. Esta directoria está fisicamente localizada no servidor de integração. De referir que os ficheiros a ser carregados têm a mesma estrutura das tabelas onde agora serão inseridos.
2. É efectuado um delete, utilizando como chave a instituição a carregar, em todas as tabelas do ODS que sejam destino de carregamento dos ficheiro.
3. A partir dos ficheiros provenientes das Instituições, é efectuado o carregamento das tabelas com a mesma nomenclatura no ODS. Este carregamento é efectuado através de mapeamento directo e sem efectuar qualquer tipo de transformação, garantindo com isto a máxima *performance* na inserção dos registos. Existe um valor acrescentado em seguir esta metodologia pois assim não precisamos de passar por um servidor para efectuarmos transformações nos dados, estes podem ser carregados a partir dos ficheiros directamente para o seu RDBMS de destino.
4. O processo subsequente tem como fonte a tabela com os registos a eliminar. Esta tabela contém as chaves a partir das quais seleccionamos os registos a eliminar. A partir destas, é feito um *update* para marcação dos registos a eliminar.
5. Os registos a eliminar marcados são agora copiados para as respectivas tabelas de histórico, que têm uma estrutura igual às de destino do processo operacional, exceptuando na inclusão do campo correspondente à data de remoção e na sua nomenclatura, onde são prefixadas por “REM_”
6. Neste passo vamos eliminar das tabelas finais os registos marcados para eliminar no passo 4, através de um processo de *delete* por chave de instituição e indicador de remoção.
7. Aplicando todas as regras de transformação e integridade é efectuado o carregamento a partir do ODS nas restantes tabelas destino. Todos os registos que não estejam em conformidade com as regras definidas, desde que não seja registado erro na validação de nulidade da sua chave primária, serão inseridos com o respectivo erro nas tabelas. Assim como, nas tabelas com o mesmo nome mas prefixadas com “ERR_”, mas aqui incluindo o código identificador do erro detectado. Após o *término* do processo e a partir das tabelas prefixadas com “ERR_” é fácil analisar problemas que tenham ocorrido no processamento.
8. Neste processo é feita uma comparação entre a informação que está pronta para sair da STA e entrar nas tabelas finais. Nesta fase, a partir da data de processamento, em cada nova inserção é actualizado o campo de data de inserção e de actualização ou *update*.
9. No final do processo e após confirmação de que o mesmo decorreu sem problemas é efectuado um processo de compressão e cópia dos ficheiros que foram fonte de processamento, para um directório no mesmo servidor que serve para manter o histórico dos ficheiros carregados. A nomenclatura dos ficheiros guardados tem como prefixo o código da



instituição à qual pertencem e, no sufixo, a data de processamento. Todo este processo é efectuado por *Shell Script*⁴Unix⁵ e invocada via ODI.

A Framework de controlo de processos foi criada com o objectivo de ser genérica para todos os processos desenvolvidos em ODI. A hierarquia pensada contempla um objecto que representa um projecto global que contem vários processos, representando os *package* ou interfaces do ODI a serem invocados. Cada processo poderá estar presente em várias execuções. Para dar suporte a esta *framework*, foram criadas as seguintes estruturas:

- CTRL_ACTION – Acções a serem despoletadas de acordo com os erros gerados;
- CTRL_DEPENDENCIES - Gere as dependências entre processos;
- CTRL_ENTITY – Informação respeitante às entidades envolvidas num projecto;
- CTRL_ERROR – Informação referente a erros gerados em cada processo;
- CTRL_EXECUTION – Controla a execução de cada processo e guarda todas as variáveis e parâmetros necessários à boa execução do mesmo;
- CTRL_EXTRACT_LOG – Log relativo a extracções de origens externas ao projecto;
- CTRL_HOLIDAYS – Informação relativa a dias de não utilização do sistema destino;
- CTRL_LOAD_LOG – Log relativo a inserções no sistema destino;
- CTRL_MSG - Controlo e criação de mensagens no processo para envio por *email* ou SMS;
- CTRL_PROCESS – Guarda todo o histórico de execução e “reexecução” de cada processo;
- CTRL_PROJECT – Informação sobre o projecto;
- CTRL_SESSION – Informação de controlo de sessões. Específico para projectos de ODI;
- CTRL_STATUS – Status do projecto;
- CTRL_STORAGE – Controlo de *storage* disponível por projecto e entidade;
- CTRL_TIME_WINDOW – Possíveis dias de execução por semana.

⁴ Linguagem scripting directamente aplicada no SO

⁵ Sistema Operativo Multitarefa e Multi-Utilizador original da Bell Labs

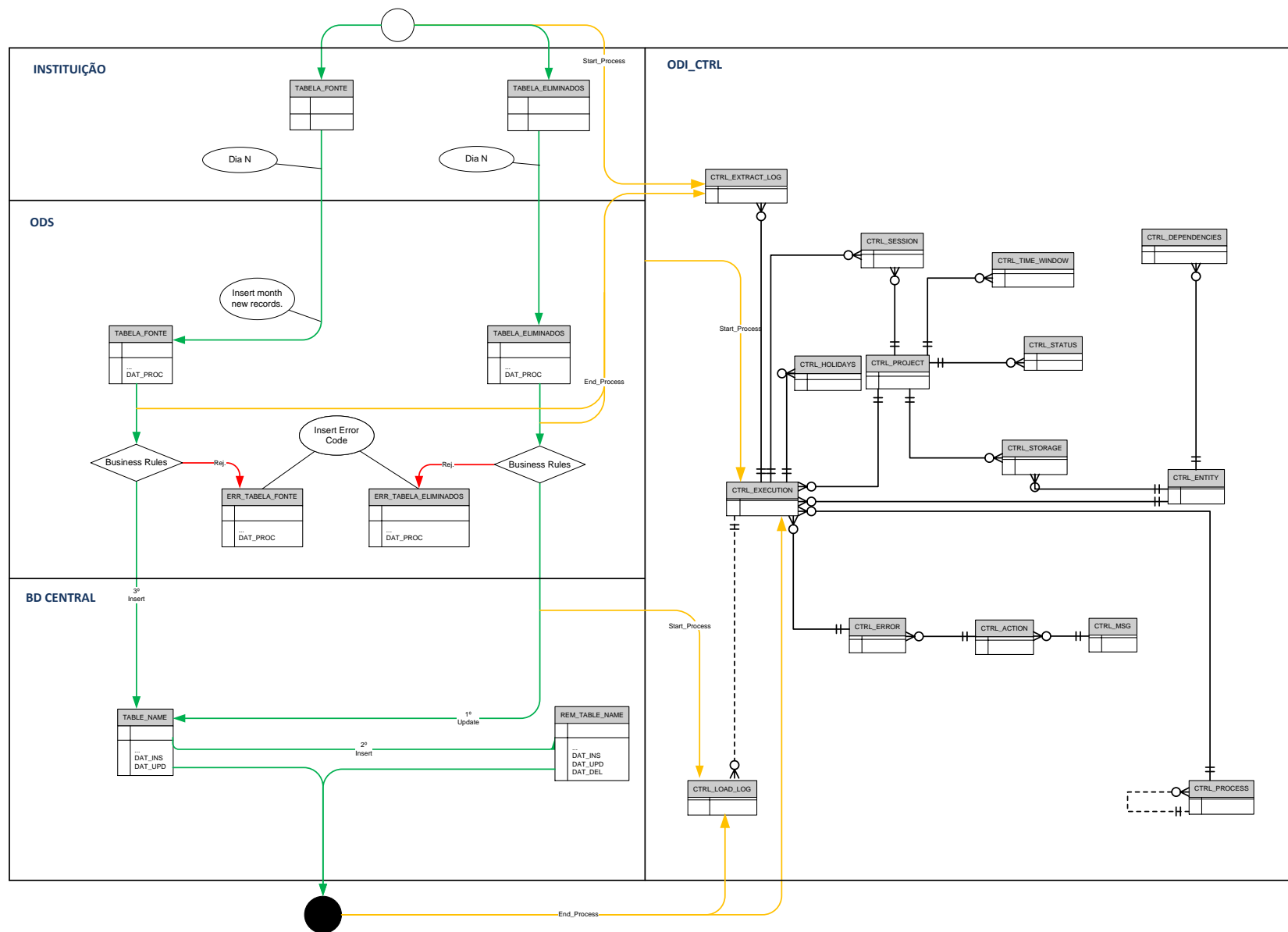


Figura 10 - Framework de carregamento a partir de RDBMS

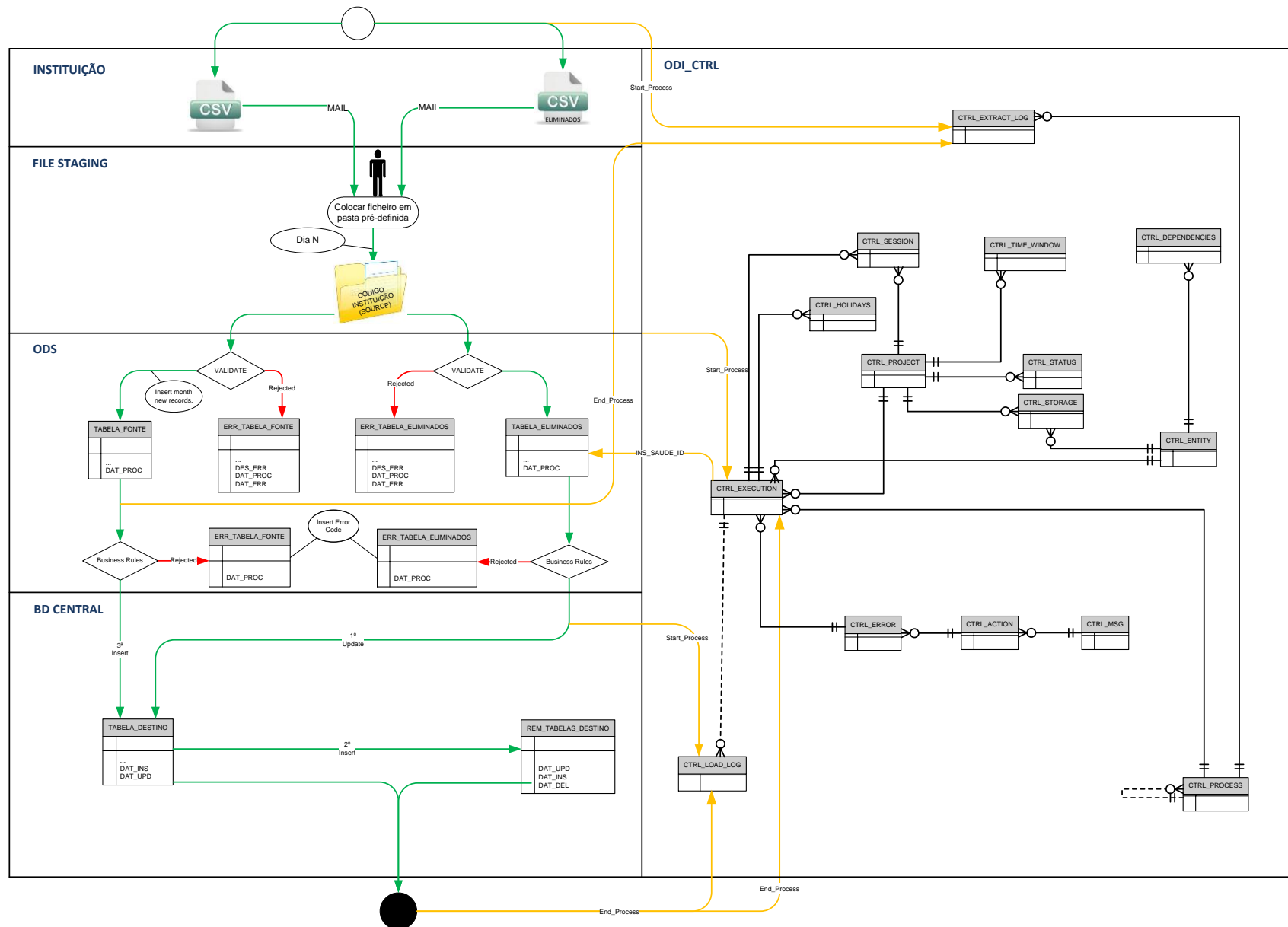


Figura 11 - Framework de carregamento a partir de ficheiros



2.1.3.7 Processos de integração

Foi criado um *package* Main com a responsabilidade de validar, entre outros, quantas entidades existem para processar, se é uma execução ou uma “reexecução” e ainda, avaliar se a integração a ser feita é por acesso directo à RBDMS ou via ficheiro.

Neste *package* foram definidas a maioria das variáveis usadas no processamento, como sejam a Data de Processamento ou a Entidade que está a ser processada. Esta informação é transportada para os seus *packages* dependentes.

Em termos de tarefas do *package* Main, posso referir:

- Avaliar quantas entidades há para processar:
 - O *package* de carregamento será executado tantas vezes quantas entidades existam, visto que as entidades são tratadas de forma sequencial. Embora fosse possível paralelizar este processamento devido a questões inerentes à própria instituição, esta opção não foi seguida.
- Avaliar se vai ser uma primeira execução ou de uma “reexecução”:
 - Para uma execução normal, as tabelas do módulo de controlo são actualizadas e os registos relativos aos vários processos são inicializados para a Entidade e Data de Processamento
 - Para reexecutar, as tabelas de controlo são actualizadas, mas não são adicionados novos registos.
- Avaliar a entidade quanto à sua origem:
 - Se existe *interface* via BD, são executados os processos criados para aceder à Base de Dados;
 - Se tratamos uma entidade que envia ficheiros, são lançados os processos preparados para aceder aos ficheiros;
 - Em ambos os casos vai existir um módulo de controlo de processos comuns;
 - Em caso de falha o módulo de controlo é actualizado.
- Por fim, será integrada a informação a partir do ODS no esquema central. Este *package* será partilhado independentemente da origem dos dados, isto porque a estrutura destino no ODS é a mesma.
 - Cada um destes interfaces tem um ID correspondente no módulo de controlo;
 - Antes da execução de determinado interface, a variável que controla o *step* onde se encontra o processo assume o ID, e o módulo de controlo é verificado de forma a determinar se para a Entidade e Data de Processamento, essa *interface* já foi executada.
 - Caso a *interface* já tenha sido executado, esta não voltará a ser e a variável V_STEP assume o ID do *interface* seguinte
 - Caso contrário, o *interface* é executado, e o módulo de controlo é actualizado
 - O *package* segue este comportamento padrão até não haver mais interfaces a executar
 - Caso seja detectada alguma falha nos diversos componentes que constituem este *package*, a execução é abortada e o facto é registado no módulo de controlo.

2.1.4 Análise de Resultados

A solução proposta e implementada coincidiu quase na sua totalidade. Relativamente ao que poderia ter ficado melhor e que não foi possível implementar, devo referir o Paralelismo de processos no ODI. Devido a um problema detectado aquando dos desenvolvimentos, que não foi possível solucionar, e que



se traduzia em quebras de ligação nas comunicações. Sempre que eram lançados dois processos em paralelo o erro de *time out* gerado pelo ODI não ajudava a despistagem da situação pois era pouco explícito. Qual a origem deste *Time Out*, foi uma pergunta para a qual não encontrei resposta. Testei várias hipóteses, inicialmente pensei que seria na BD destino, mas os processos do ODI quando ocorria esta falha nem chegavam a BD. Confirmamos e validamos esta situação numa tabela de sistema do Oracle, a *V\$Session*. Pensámos que poderia ser algum problema nos agentes que eram colocados nas instituições e sem os quais não seria possível ligarmos às fontes. Visto que os agentes estavam em execução e conseguíamos conectar-nos a estes sem problemas, fiquei sem pistas para seguir e pensei que o problema poderia estar no sistema de gestão da rede, embora sem nunca ter a certeza absoluta. Aquilo que parecia, era que, por alguma razão, quando o ODI lançava dois processos em paralelo, em determinado ponto perdia a ligação a um deles e este não retornava qualquer erro e assim, o servidor de integração nunca reconhecia a ocorrência deste até acontecer o *time out*.

Todos os restantes pressupostos foram cumpridos e a solução revelou-se uma mais-valia para a ACSS.

2.2 Projecto de criação de modelo analítico na ACSS

2.2.1 Contexto

Este projecto na ACSS teve uma complexidade elevada não só devido ao número de componentes envolvidas mas principalmente devido à especificidade de cada uma.

Este consistiu na criação de um modelo analítico criado a partir de DW sobre o qual a partir de uma ferramenta de exploração de dados se geraram KPIs, relatórios, documentos e *dashboards*. Antes do projecto só era possível criar análises relacionais, a pedido, feitas directamente sobre o modelo transaccional, que não estava preparado para dar resposta à maioria das necessidades dos responsáveis.

2.2.2 Requisitos e Motivação

Este projecto teve como grande motivação o facto de poder disponibilizar um conjunto de informação de gestão, crítica para a organização, de forma automática e com o mínimo de intervenção humana possível.

Os automatismos permitiram aumentar a produtividade, libertando os *key users* de tarefas rotineiras e consumidoras de tempo na produção de informação, dando-lhes a possibilidade de se focarem mais na análise da mesma e assim melhorar a qualidade dos processos de tomada de decisão.

2.2.3 Solução Técnica e Execução do Projecto

Tendo como ponto de partida que o projecto iria consistir na geração de um novo modelo analítico e utilizamos na sua construção as tecnologias ODI e OBIEE, iniciámos o desenvolvimento com reuniões com os *key users* para definição de requisitos.



Nesta primeira fase introduzimos uma *framework* na criação de documentação por forma a facilitar a manutenção os objectos criados futuramente. Só desta forma foi possível após o fecho do projecto conseguir fazer manutenção deste sem a presença de elementos que o conheçam profundamente. Sempre defendi esta metodologia, devido à experiência prática que adquiri, e que me mostrou inúmeras vezes as dificuldades inerentes ao facto conhecer uma solução, unicamente através da sua utilização, quando não existia qualquer informação adicional sobre a mesma. Em termos de *time consuming*, esta nunca será uma boa solução.

Assim, a cada vez maior rigidez nas boas práticas e normas que certificam o trabalho desenvolvido na área da consultoria, tal como os modelos CMMI [16] ou ITIL [17], que contemplam um número de pressupostos que em inúmeras consultoras são tidos como obrigatórios devido os seu nível de certificação obrigam-nos no desenvolvimento do nosso trabalho a adoptar metodologias que se traduzem na utilização das melhores *best practices* do mercado.

Esta questão é muito pertinente, e com maior expressão ainda, quando colocada na realidade do mercado Português, onde o dinheiro não abunda. Isto porque os níveis de rentabilidade dos projectos muitas vezes obrigam as consultoras a não poderem subir o seu nível de certificação, devido a quantidade de pressupostos que lhe são colocados para entregar um projecto com o nível de certificação que já foi alcançado.

Em termos de mercado internacional esta situação é exactamente a oposta, ou seja, se os níveis de certificação não forem suficientemente elevados provavelmente representarão um factor eliminatório num concurso internacional. Assim é necessário racionalizar este tipo de opção tendo em conta os vários cenários.

Posto isto, na altura da definição dos requisitos, desenvolvi um modelo documental para registar toda a informação sobre os indicadores e seus derivados criados durante o projecto e onde pudessem ser registados os criados futuramente. A informação aqui introduzida seria a título de exemplo a data de criação, autor, versão, forma de cálculo, variantes, etc..

Desenvolvi também um modelo de planeamento para *dashboards*. Foram desenhados os relatórios que foram adicionados tal como, a forma como seriam dispostos. Mais uma vez tratou-se de uma problemática nova para mim, sendo de realçar que este foi o meu primeiro projecto de desenvolvimento de *dashboards*. Conclui que estes obedecem a regras no seu desenho muito semelhantes a páginas web, em que a estrutura visual é fundamental para o seu sucesso. É importante que sejam fáceis de interpretar e apelativos. Actualmente esta componente passou a ser fundamental nos projectos de BI, e neste caso também o foi.

A definição de *dashboarding* é a forma de disponibilizar visualmente informação objectiva, consolidada e estruturada, que permita uma fácil e rápida interpretação. Existe um conjunto de *best practices* que foram seguidas no desenvolvimento do projecto e que tomei conhecimento na referência bibliográfica [18], às quais faço referência em seguida:

- Disponibilizar a informação directamente relacionada num único ecrã, ou seja, evitar partir a informação por várias páginas;
- Evitar a necessidade de *scrolling*;
- Contextualizar a informação disponibilizada;
- Incluir factores de comparação e sugerir acções na visualização dos indicadores;
- Utilizar escalas adequadas, que devem dar uma perspectiva real das quantidades apresentadas e não podem iludir os utilizadores;
- Utilizar níveis de precisão adequados nos indicadores, pois evita perdas de tempo com leituras e interpretações de informação desnecessárias e pouco relevantes;



- Escolher os indicadores mais adequados, que facilitem e acelerem a interpretação da informação disponibilizada;
- Escolher soluções gráficas flexíveis e adequadas que facilitem e acelerem a interpretação da informação disponibilizada;
- Uniformizar a leitura ao longo do *dashboard*:
 - Aplicar as mesmas cores para os mesmos indicadores;
 - Aplicar os mesmos tipos de gráficos para os mesmos tipos de comparação;
 - Aplicar os mesmos tipos de alertas;
- Facilitar a interpretação da informação disponível para acelerar a sua leitura. Por exemplo, evitar cores berrantes, muito próximas, muito apagadas ou um número muito elevado de cores;
- Apresentar a informação de forma equilibrada, dado que o espaço utilizado num *dashboard* desce de importância do canto superior esquerdo para o canto inferior direito, e por esta razão, a informação que se destaca na visualização deverá ser a mais importante;
- Os títulos não devem ser mais apelativos que os indicadores;
- Destacar a informação mais importante e não cair no erro de chamar a atenção para tudo;
- Aproveitar bem o espaço disponível, ou seja evitar decorações desnecessárias e ainda evitar soluções de pesada implementação para responder a pormenores visuais;
- Utilizar cores de forma ponderada, ou seja, utilizar cores apelativas apenas para a informação mais importante, podendo utilizar contrastes;
- Manter as cores para os mesmos indicadores ao longo do *dashboard* ou para o mesmo tipo de indicador associado;
- Podem ser utilizadas figuras geométricas para além das cores, tais como o círculo, triângulo ou quadrado como forma de ajudar utilizadores que sofram de daltonismo;
- Criar uma apresentação apelativa, baseando-se na nossa intuição e naquilo que consideramos que a maioria das pessoas aceita e tolera positivamente.

Feita esta análise, gerei um documento de especificação funcional e iniciámos a análise técnica. Nesta fase era preciso estruturar inúmeras componentes do projecto e aqui foi especialmente útil a experiência que tinha em projectos de integração e *reporting* em que estive envolvido ao longo da minha carreira.

A primeira preocupação seria a modelação de todo o DW uma vez que somente após terminada esta componente do projecto poderíamos estruturar as interfaces de carregamento das estruturas criadas. Conhecíamos as fontes disponíveis e era preciso desenhar um modelo que desse resposta ao conjunto de indicadores e relatórios requeridos na fase de análise. A leitura das publicações de Ralph Kimball [19] serviram-me de fonte de conhecimento para muitas situações de projectos de DW. Uma delas consiste no facto de a realidade num DW, vista hoje, ou num qualquer período futuro, ter de ser obrigatoriamente igual. Neste projecto tal não se verificou, pois o facto da informação operacional chegar por vezes em datas posteriores à do carregamento, não permitiu disponibilizar um histórico de factos. Com esta realidade o nosso modelo deixou de ser um DW puro, tendo sido necessário estruturá-lo, bem como às suas interfaces de carregamento, por forma a dar suporte a esta situação.

2.2.3.1 Infra-estrutura

Neste projecto a infra-estrutura em termos de Hardware, tal como já acontecia no projecto de integração operacional descrito anteriormente, era-nos desconhecida enquanto fornecedores. Por esta razão, apenas posso apresentá-la ao nível do Software disponível. No total foram-nos disponibilizadas 4 máquinas cujo modelo pode ser visualizado na figura seguinte.

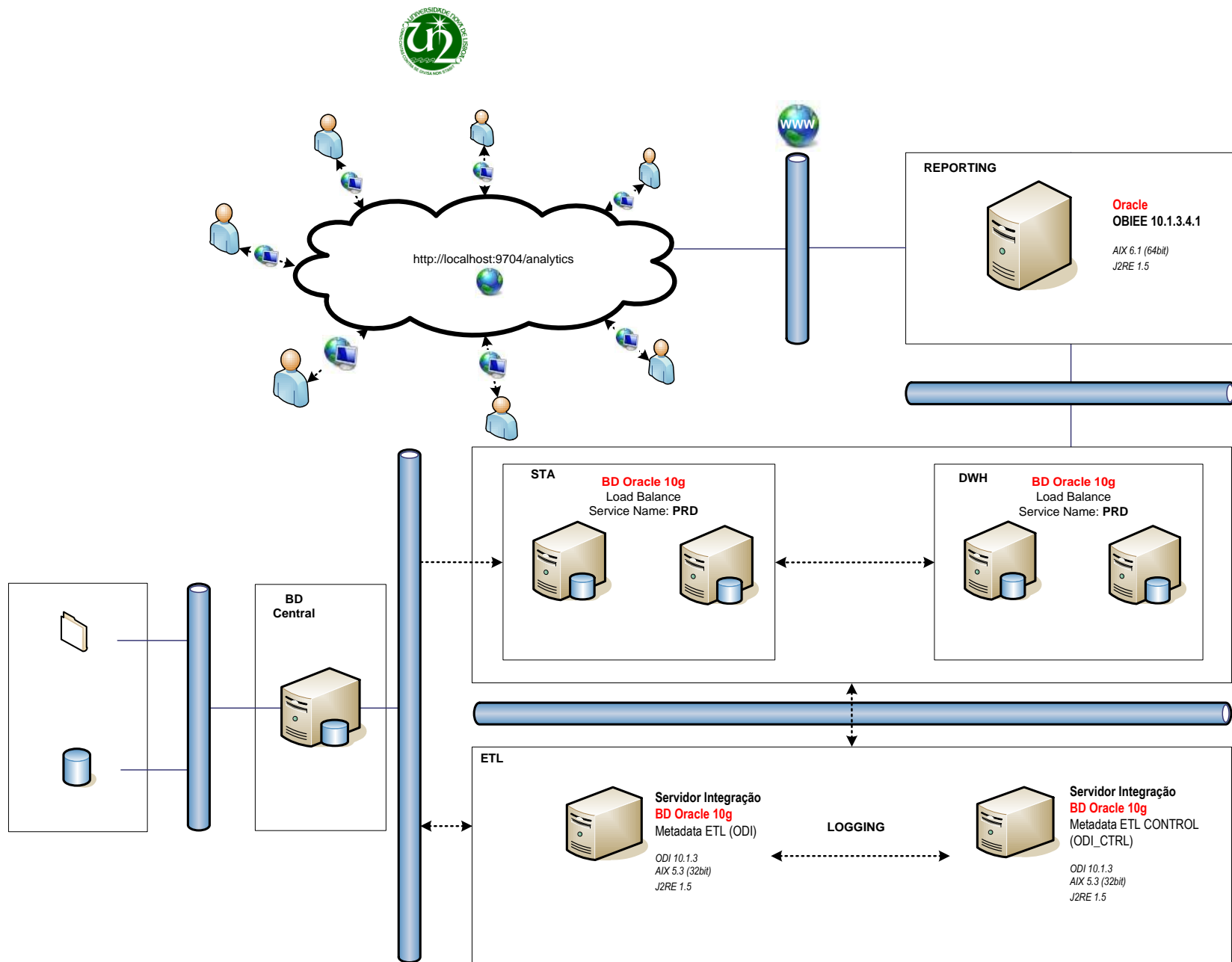


Figura 12 – Infra-estrutura da Solução do Modelo Analítico



2.2.3.2 Arquitectura Macro e modelo físico

Na definição da arquitectura, utilizei um modelo que já me era conhecido de outros projectos de carregamento de DW em que participei. Pude comprovar que este modelo facilita a integração, na criação dos processos, pois organiza de forma visível a sua estrutura. Este modelo permite ainda a não interferência na *performance* dos repositórios que lhe servem de fonte, e que deles necessitam para dar resposta a aplicações que sobre eles actuam.

Assim, em alto nível, podemos visualizar o modelo criado na figura seguinte:

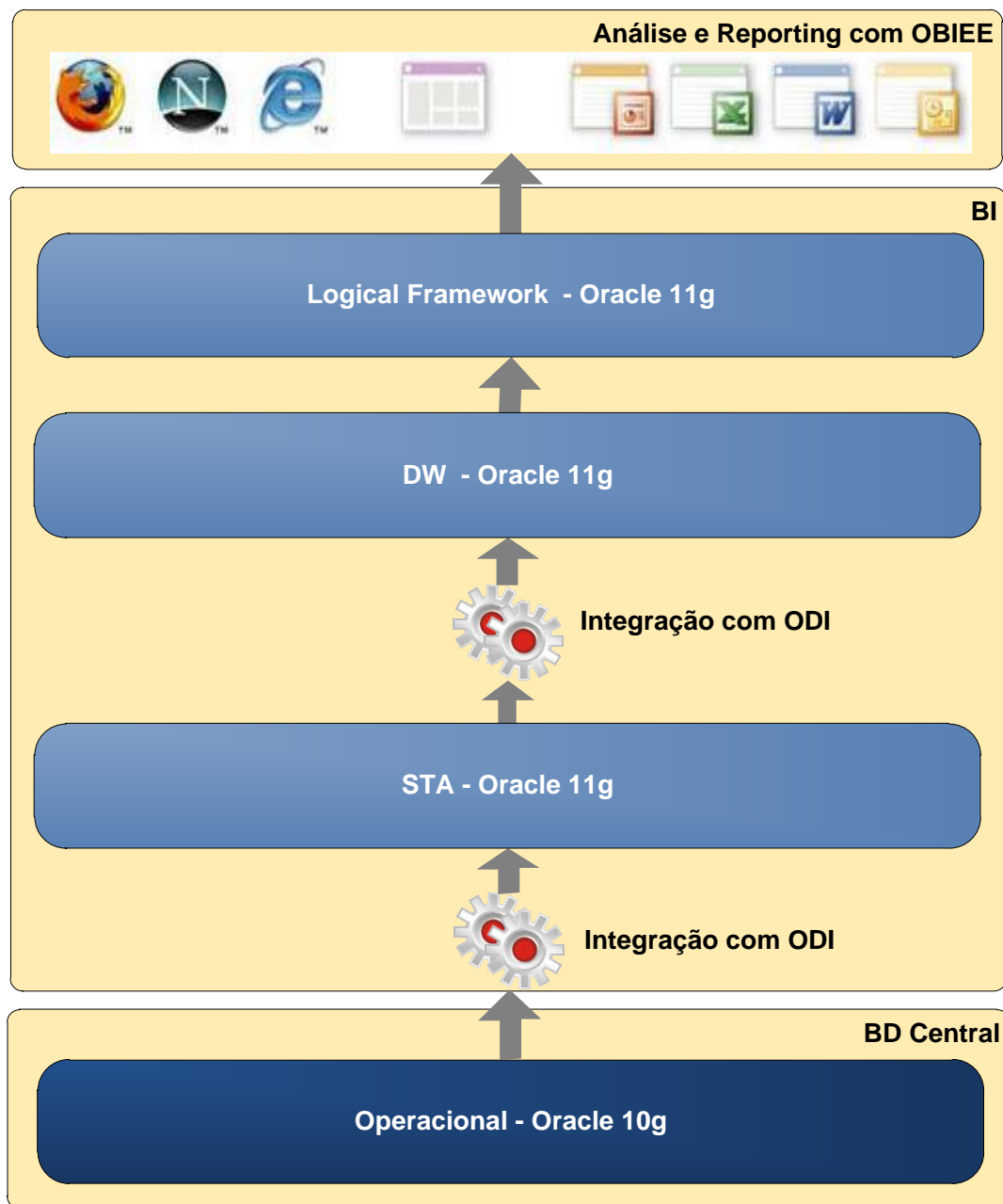


Figura 13 – Arquitectura Macro do Modelo Analítico



A arquitectura física consistiu numa componente de STA, um DW, e uma *Logical Framework*. Todas as componentes são utilizadores de BD que pertencem ao mesmo repositório físico, uma RDBMS Oracle 11G. Todas as interfaces entre a fonte, o modelo operacional, a STA e o DW, serão carregadas e modeladas utilizando o ODI. A ferramenta além de gerar as interfaces também gere todo o processo de carregamento, a geração de *logs* e controlo de erros. A camada lógica denominada *Logical Framework* será desenvolvida através da criação de *views*⁶ PL/SQL Oracle que assentam na sua totalidade sobre o modelo físico de DW.

Separando a solução em fases distintas, sequenciais em termos temporais, temos:

1. Utilizando o ODI, extracção da informação do modelo operacional para a STA;
2. Com o ODI, construção do processo de integração que vai ler da STA para o DW;
3. Após a integração no modelo físico do DW, construção da *Logical Framework*;
4. Todo o modelo gerado na fase anterior é utilizado como fonte da ferramenta analítica, o OBIEE;
5. Ferramenta de *reporting*, com acesso através de cliente Web para exploração.

Funcionalidade de cada componente:

- Staging Area – Tem como finalidade fazer a transição da informação entre o modelo OLTP e o OLAP, ou seja, entre o operacional e o DW. Na STA temos a informação a ser integrada num delta do processamento mensal. É ainda nesta camada que toda a informação é estruturada no mesmo formato que o DW, a diferença reside no facto de não conter SK [20].
- DW – A forma de modelar esta camada foi recorrendo ao *Snowflake*, que é uma derivação do *Star Schema*, divulgada por Ralph Kimball. Este tipo de modelação da informação permite hierarquizar as dimensões existentes não funcionando de forma diferente do Star Schema em relação às tabelas de factos. É muito utilizada em modelos analíticos como DW ou DM [22]. O modelo continha tabelas de factos, tabelas de dimensão hierarquizadas e tabelas agregadoras. Para além das referidas, e para dar resposta ao modelo de negócio, durante o projecto optei pela criação de estruturas adicionais, tabelas agregadoras com indicadores pré-calculados e tabelas de relação como auxiliares.

Esta forma de modelar definida no projecto era-me conhecida de outros ambientes de DW. Neste momento considero que é a forma mais simples e funcional de modelar, no que diz respeito à construção de um sistema de raiz. Apresenta também mais-valias inequívocas, no desenho, pois pode ser totalmente orientado ao modelo de negócio, e, principalmente na manutenção das interfaces e estruturas.

Defini como necessárias dimensões de tipo 0, 1 e 2. Existem ainda as de tipo 3, 4 e 6 que não utilizei neste projecto. Vou explicar de forma breve, o tipo de dimensões existentes e o seu funcionamento:

- Tipo 0, são as que ao longo do tempo de vida da solução nunca são alteradas, ou seja não há inserções ou alterações;
- Tipo 1, onde não existe manutenção de histórico, funcionando apenas com base em actualizações e inserções;
- Tipo 2, onde tem lugar uma gestão de histórico, para tal, incluímos nas respectivas tabelas, campos, que por defeito se referem à validade dos registos, tais como, a data de início e fim. Ou seja, um registo é considerado activo dependendo da data em que efectuamos a pesquisa. Quanto aos que consideramos activos à data actual, vão conter como valor por defeito na sua data de fim, um *default* do tipo data, atribuído manualmente, e que, não prevemos que seja alcançado durante o tempo de vida útil do sistema, já que assim, conseguimos manter a integridade do modelo.

⁶ Tabela lógica baseada em uma ou mais tabelas ou views.



- Tipo 3, têm um comportamento semelhante às de tipo 2, com a diferença que o histórico neste caso só é mantido em relação ao seu estado original. As evoluções são registadas por *updates* sempre na mesma linha, não sendo inserida uma nova linha por cada alteração.
- Tipo 4, o comportamento também é semelhante às do tipo 2, mas aqui temos uma tabela de histórico adicional, registamos as alterações que a dimensão for sofrendo. Estas são guardadas em cada nova iteração mantendo na tabela de dimensão apenas o registo mais actual.
- Tipo 6, referem-se a uma abordagem de Ralph Kimball [19], que as considera como dimensões com alterações inesperadas. Trata-se de um misto entre as de tipo 1, 2 e 3 com a diferença que, é incluído um campo, *flag* que corresponde à validade ou não do registo. Apenas um registo fica marcado como activo e todos os restantes como inactivos.

Nas tabelas de dimensão, a chave é um atributo fundamental. No modelo definido todas as dimensões tinham um código unívoco de identificação para cada registo. Este código deve ser preferencialmente numérico por razões relacionadas com *storage* disponível e *performance*. Assim, a sua chave primária será composta por um valor inteiro atribuído sequencialmente, e por essa razão, será sempre um atributo único. Neste caso a aposta numa chave de apenas 32 bits acelera o acesso a um grande volume de dados armazenados. Estas chaves serão utilizadas para possibilitar a ligação entre tabelas de dimensão e as de factos.

Assim, sempre que se justificou, isto é quando se tratavam de dimensões com volumes acima de 10 registos, ou no caso da sua chave original não ser do tipo inteiro, gerávamos uma SK.

Existem várias abordagens seguidas para criar as SK, designadamente a utilização de um objecto *sequence* da SGBD Oracle, isto é a geração de um *autonumber* na BD. Optámos pela manutenção numa tabela com uma entrada por cada nova SK segmentada pela dimensão à qual se referia, contendo assim como campos o nome da tabela e a SK disponível. Assim sempre que havia uma nova entrada numa dimensão, o processo seguido era a consulta na tabela que continha as SK, da entrada referente a esta, retornando a chave corrente da mesma. No novo registo era usado o valor sequencialmente acima do registado e posteriormente actualizado aquele que se encontrava na tabela de SK para o seu valor actual.

Existem várias razões para se utilizar as SK ou chaves substitutas, nomeadamente:

- O facto de conseguir manter o DW o mais isolado possível das regras operacionais, e assim, sempre que necessário, gerar, actualizar, remover, reciclar e reutilizar os códigos provenientes dos sistemas transaccionais.
- Num DW, no caso de uma integração de dados, a informação é mantida durante muito tempo e não pode ficar vulnerável a problemas de sobreposição de chaves.
- Desempenho no acesso à informação. Os códigos utilizados em sistemas transaccionais constituídos por uma cadeia de caracteres alfanuméricos apresentam uma deficiente *performance* no acesso à base de dados relativamente às SK, que utilizam o menor inteiro possível.
- Nas alterações de atributos nos sistemas transaccionais. As chaves genéricas serão a base para mantermos o histórico das alterações no DW. No caso do modelo carregado, foram geradas SK em praticamente todas as dimensões que tinham origem no Sistema Centralizado.
- Todas as dimensões que tenham este conceito implícito, vão ter sempre três chaves ou SK por defeito e iguais em todas as dimensões. Serão as chaves referentes ao código inexistente ou valorizado a *null* (-1), chave inexistente para determinado código (-2), e ainda, a chave correspondente a um código inválido onde se inclui códigos mal formatados (-3).

Apresento em seguida uma dimensão que representou um caso particular neste projecto, a instituição. Esta dimensão, do tipo 2, era mandatária na quase totalidade da informação, e influenciava a definição dos processos de carregamento.

A necessidade de efectuar análises históricas, em que é considerada a realidade da instituição no período de análise, despoletou este processo.

Ao contrário das dimensões de tipo 1, onde a sua chave primária é sua SK, aqui temos uma chave primária composta, constituída pela SK e a data de fecho do registo. Assim para validar se uma instituição se encontra activa em determinada data, teríamos que validar se este campo continha o valor *default* escolhido para data máxima.

Sempre que existia uma nova entrada para a mesma SK, este valor *default* alterava-se para a data de processamento corrente, o registo existente era considerado fechado e era inserido um novo, com o valor *default* na data de fecho.

O carregamento desta dimensão no formato descrito, afecta directamente o carregamento dos factos. Ou seja, no histórico de uma entidade existia uma situação que obrigava a manipular os factos a serem carregados.

Esta situação ocorria quando uma instituição era integrada numa nova ou existente mas mantendo em funcionamento os seus sistemas. Embora a instituição tivesse uma nova designação, o facto de a informação provir da mesma fonte obrigava à conversão do código dessa instituição, para o novo, onde havia sido integrada.

A situação descrita gerou aquando do carregamento do DW, a criação de uma função que validava o registo para as tabelas de factos, o verdadeiro código antes de este ser inserido.

Na escolha da ferramenta de BI, o OBIEE, o facto de no ano de 2009, quando decorreu o projecto, a Oracle estar integrada no grupo de líderes do “Quadrante Mágico da Gartner para Ferramentas de BI” [23], justificou esta escolha, confirmando-se ainda esta boa opção, ao atingir a liderança em 2010 [24].

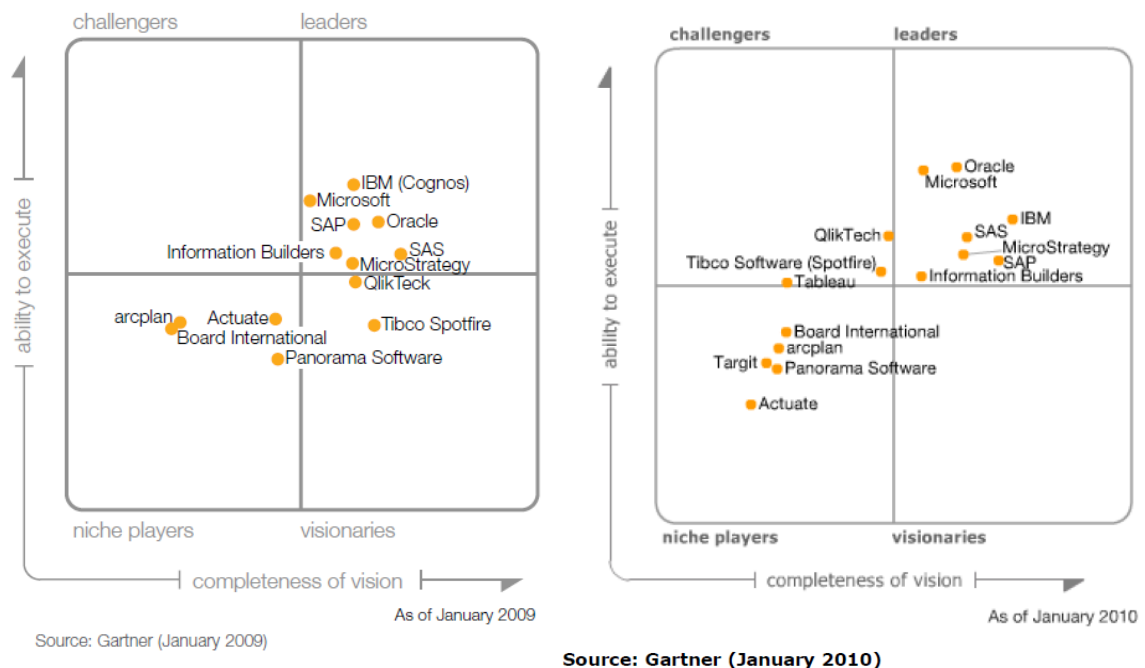


Figura 14 - Quadrante Mágico de BI do Gartner Group para 2009 e 2010



Todas as tabelas construídas tinham campos por defeito, tais como a data de inserção dos registos e a sua possível data de actualização.

Existia uma tabela que era um caso particular, pois tratava-se de uma dimensão que se ligava a todas as tabelas de factos existentes. Por ser possível efectuar remoção da informação do DW posteriormente ao seu carregamento, este não podia ser considerado um DW puro, e por conseguinte, a referida dimensão apresentava um campo indicador de remoção. Trata-se de uma *flag*, que tinha um valor *default* 0, e que no caso de o registo ser removido na sua fonte, seria actualizado com o valor 1. Pelo facto da referida tabela centralizar todos os factos existentes no DW, permitia que nas *views* ao nível da camada lógica através do código PL/SQL fosse filtrada toda a informação para que só fosse visível no modelo OLAP, todos os registos cujo indicador de remoção tivesse o valor 0.

De forma a dar resposta à criação de relatórios mais complexos, e em virtude de num sistema analítico a *performance* ser um factor fundamental, foram criadas estruturas para agregar informação de várias fontes do próprio DW, as tabelas agregadoras. Pelo facto de apresentarem especificidades que tornam o seu cálculo por via da ferramenta analítica muito complexo e pesado, esta foi a opção mais valida para solucionar este tipo de situações.

Criámos ainda tabelas auxiliares especificamente para dar resposta a um conjunto de *reports* específicos. O objectivo seria aumentar a *performance* das análises.

Quanto ao dimensionamento do DW, foi efectuado um cálculo do valor necessário em termos de *storage* a ser atribuído aos *tablespaces* disponibilizados para este utilizador. Na determinação do seu valor, foi seguida a seguinte metodologia.

- Dimensões - Foi efectuada uma contagem do número de registos na fonte, em cada dimensão. Calculou-se para cada dimensão, a ocupação de uma linha “populada”, O valor determinado era multiplicado pelo número de linhas da sua fonte. Por fim, para converter o valor da escala origem (Bytes) para a final, MB, foi feita a sua divisão duas vezes consecutivas por 1024. No casos de tabelas com valores de ocupação que considere baixos fiz uma ponderação, arredondando estes para o seu superior na mesma escala.
- Factos - Foi calculada a ocupação por linha em cada tabela de factos e o número médio dos restantes factos associados à dimensão por cada linha. Com o número médio, multipliquei o mesmo pelo número de linhas e pela ocupação de cada uma. Mais uma vez, para atingir a escala destino, MB, efectuei duas vezes consecutivas a divisão por 1024.
- Tabelas agregadoras e Auxiliares – Devido à sua dimensão ser marginal, não foram consideradas estas tabelas no cálculo do dimensionamento.

Relativamente ao particionamento, vão existir algumas tabelas que devido a sua dimensão foram particionadas. A decisão sobre qual a melhor chave para este particionamento recaiu, em todas, sobre o ano e a instituição.

A razão desta decisão deveu-se a questões de *performance*, uma tabela particionada é uma tabela lógica, isto significa que é vista na BD pelo *user* como uma tabela única, mas em termos físicos para o RDBMS, consiste em tantas tabelas, quantas o número de partições que sejam criadas a partir da mesma. Isto permite por exemplo alocar partições em *tablespaces* ou até *storages* diferentes, facilitando a sua gestão e incremento de *performance*.

Quanto ao processo de **Backup**, foi definido um *Cold Backup*⁷ Mensal por *tablespace*. Este deve ocorrer dois dias antes do processamento mensal, sendo mantidas imagens dos 2 meses anteriores.

⁷ Cold backup ou offline, Com a BD em modo offline é feito um backup evitando que os dados possam sofrer actualizações durante o processo de backup. O downtime da BD neste processo é extenso.



- Carregamento inicial - Pelo facto de serem gerados deltas para integrar a informação que já existia anteriormente no modelo OLTP que serviu de fonte. Foi necessário efectuar um *update* ao nível do user de BD do OLTP no campo referente à data de entrada da informação nas tabelas fonte, e assim permitir a geração dos deltas para carregamento da informação. Como fonte para o referido *update*, o campo utilizado seria a data do próprio evento ao qual se referisse. Após a passagem a produção, este campo passou a comportar o valor referente a data de processamento para serem identificados os novos registos e integrados no novo delta, independentemente da data a qual o evento se refere.

Aquando da passagem a PRD, e efectuado o primeiro carregamento, garantimos o carregamento prévio de todas as dimensões manualmente. Caso esta situação não se verificasse, todas as tabelas que se relacionem com estas, ficariam sem referências, ou seja sem a chave estrangeira que referencia as mesmas.

- Processo de Integração – O processo de integração não será automático no carregamento de todas as estruturas de dados. Algumas dimensões são mantidas através de processos manuais. Neste caso, as que contêm apenas descritivos para os códigos apresentados e cuja manutenção acontece apenas esporadicamente e por razões específicas. No modelo criado, isto aconteceu por exemplo, com a hierarquia do tempo, do âmbito de análise, dos grupos etários e do género, sendo informação base na ferramenta analítica que só pode ser manipulada em situações específicas.

Existiam também dimensões, que não sendo de carregamento manual, pelo facto dos seus descritivos serem atribuídos manualmente e no caso de surgir algum código novo, este era integrado sem a respectiva descrição, *ie*, com uma descrição por defeito e igual para todas as tabelas de dimensão.

Quando este registo era canalizado na mesma operação para a tabela de erros do processo, permitia que estas situações fossem detectadas rapidamente e assim de forma manual adicionadas as respectivas descrições.

O fluxo mensal de carregamento garante a actualização do modelo. Este processo irá ser executado todos os meses, e após a integração no OLTP que é a sua fonte. Este processo é modular ao nível de granularidade mais baixo, e assim, em caso de falha em qualquer ponto do mesmo, pode ser reexecutado a partir de um ponto específico anterior a este, e não desde o primeiro *step*. Os pontos de ré execução foram especificados de acordo com a sequência do processo e das dependências existentes. Significa que para cada interface antes de ser executada o processo vai verificar se há condições em concordância com a interface precedente, em caso de falha, esta será reexecutada para dar condição de arranque á seu posterior.

○ Framework de carregamento

Todos os processos de carregamento terão como base uma *framework* específica que irá reger todo o fluxo de integração. Assim, com base na experiência adquirida em projectos anteriores, e após alguma investigação descobri, numa pesquisa na Web utilizando o Google, uma *framework* que apresentava soluções específicas para carregamento de DW, embora fosse específica para OWB [25]. Adaptei a mesma para a realidade do ODI e construí assim uma versão para carregamento de tabelas de dimensão e outra para carregamento de tabelas de factos.

A *framework* de carregamento teve adicionalmente de incluir a *framework* de controlo de processos. É através desta que é feito o controlo do fluxo dos vários processos e as suas interdependências. Sendo genérica para todos os processos e disponível numa BD onde se encontra toda a *metadata* do ODI.

▪ Framework de Carregamento de Dimensões

Esta *framework* definida em 4 passos tem como objectivo efectuar a inserção numa tabela destino no modelo de DW. O seu esquema pode ser visualizado na figura 15.



1. É feito um *truncate* a todas as tabelas que serão utilizadas no utilizador de *staging* sufixadas com 300_DLT e 400_CHG. O significado destes sufixos será no caso das 300 a geração de um delta de carregamento e nas 400 a geração de um conteúdo com formato igual à tabela final no DW.
 2. A partir da BD (OLTP) central será efectuado uma leitura de todos os campos necessários das fontes de cada uma das dimensões. Esta informação é carregada directamente, sem efectuar qualquer tipo de transformação. O destino deste processo são as tabelas sufixadas por 300_DLT.
 3. Aplicando todas as regras de transformação e integridade será efectuado o carregamento a partir das tabelas com sufixo 300_DLT para as 400_CHG. Todos os registos que não estejam em conformidade com as regras definidas, desde que não seja registado erro na validação de nulidade são inseridos com o respectivo erro nas tabelas 400_CHG. Os registos em que sejam detectados erros, são inseridos nas 450_ERR. Após o *término* do processo e a partir das tabelas sufixadas com 450_ERR é possível analisar problemas que tenham sido gerados no processamento e verificar a razão pela qual os registos mostram algum tipo de incongruência.
 4. Será feita uma comparação entre a informação que está pronta para sair da STA e entrar no DW e aquela que já existe dentro da tabela final do DW, a dimensão. Assim, após esta validação estamos prontos para uma correcta inserção ou actualização dos registos. Em cada nova inserção será preenchido o campo que contém a data de inserção e em cada nova actualização o campo referente a data de actualização ou *update*.
- Framework de Carregamento de Factos
- A *framework* definida em 5 passos efectua a inserção na tabela destino no modelo de DW, esta pode ser visualizado na figura 16.
1. É feito um *truncate* a todas as tabelas que serão utilizadas no utilizador de STA sufixadas com 300_DLT, 300_DEL_DLT, 400_CHG e 400_DEL_CHG. O significado destes sufixos tal como nas dimensões, será no caso das 300_DLT a geração de um delta de carregamento, na 400_CHG a geração de um conteúdo com mesmo formato da tabela final do DW, na 300_DEL_DLT e 400_DEL_CHG seguindo a mesma lógica em termos de formato mas com a informação a ser marcada como eliminada no DW.
 2. A partir da RDBMS (OLTP) central será efectuado um carregamento directo de todos os campos necessários das respectivas fontes. Este carregamento será efectuado através de uma leitura e escrita directa sem efectuar qualquer tipo de transformação. O destino deste processo serão as tabelas sufixadas por 300_DLT.
 3. Aplicando todas as regras de transformação e integridade será efectuado o carregamento a partir das tabelas com sufixo 300_DLT para as 400_CHG. Todos os registos que não estejam em conformidade com as regras definidas, desde que não seja registado erro na validação de nulidade da sua chave, serão inseridos com o respectivo erro nas tabelas 400, assim como nas 450 com o código de erro respectivo. Após o *término* do processo e a partir das tabelas sufixadas com 450_ERR será possível analisar problemas que tenham sido gerados no processamento e verificar a razão pela qual os registos mostraram algum tipo de incongruência relativamente ao esperado.
 4. Serão marcados como registos para eliminar, os que se encontram na tabela factual final, quando comparada com a tabela sufixada com 400_DEL_CHG.
 5. Os registos na factual no DW serão inseridos ou actualizados a partir da tabela 400_CHG. Numa inserção será “populado” o campo referente à data de inserção e, em cada nova actualização o campo referente à data de actualização ou *update*.



- A Framework de controlo de processos
Esta *framework* é a mesma que foi utilizada no projecto anterior, de integração do sistema operacional e encontra-se descrita no capítulo 2.1.3.6.
- ODI - Package de lançamento da integração
Foi criado um primeiro *package* com a responsabilidade de garantir a sequencialidade do processo, onde as Dimensões são carregadas antes dos Factos. Assim, sempre que este é lançado, vai verificar se a execução das Dimensões já ocorreu para aquela data de processamento.
O *package* seguinte irá validar, quantas entidades existem para processar, e se, ocorrerá uma execução ou “reexecução”, para além de avaliar se a integração está a ser feita para Dimensões ou Factos.
Em termos de tarefas, encontram-se as seguintes:
 - Avaliar quantas entidades há a tratar:
 - O *package* irá ser executado tantas vezes quantas entidades existam, visto que as entidades são tratadas de forma sequencial.
 - Avaliar se é execução normal ou “reexecução”:
 - Para uma execução normal, as tabelas do módulo de controlo são actualizadas e os registos relativos aos vários processos são inicializados para a entidade e data de processamento;
 - Para uma “reexecução”, as tabelas de controlo são actualizadas, mas não são adicionados novos registos nestas.
 - Avaliar o modo de execução:
 - Se vamos executar o processo das Dimensões;
 - Se vamos executar o processo dos Factos.
 - Quando a execução de uma entidade termina, independentemente de ser com sucesso ou insucesso, é iniciada a integração da seguinte, até não existirem mais entidades a tratar. Em caso de falha o módulo de controlo é actualizado.
 - Após o *término* da integração dos dados da última entidade, é executado um procedimento que actualiza as *materialized views* e carrega as tabelas agregadoras e, só após estas, o *package* principal termina a sua execução.

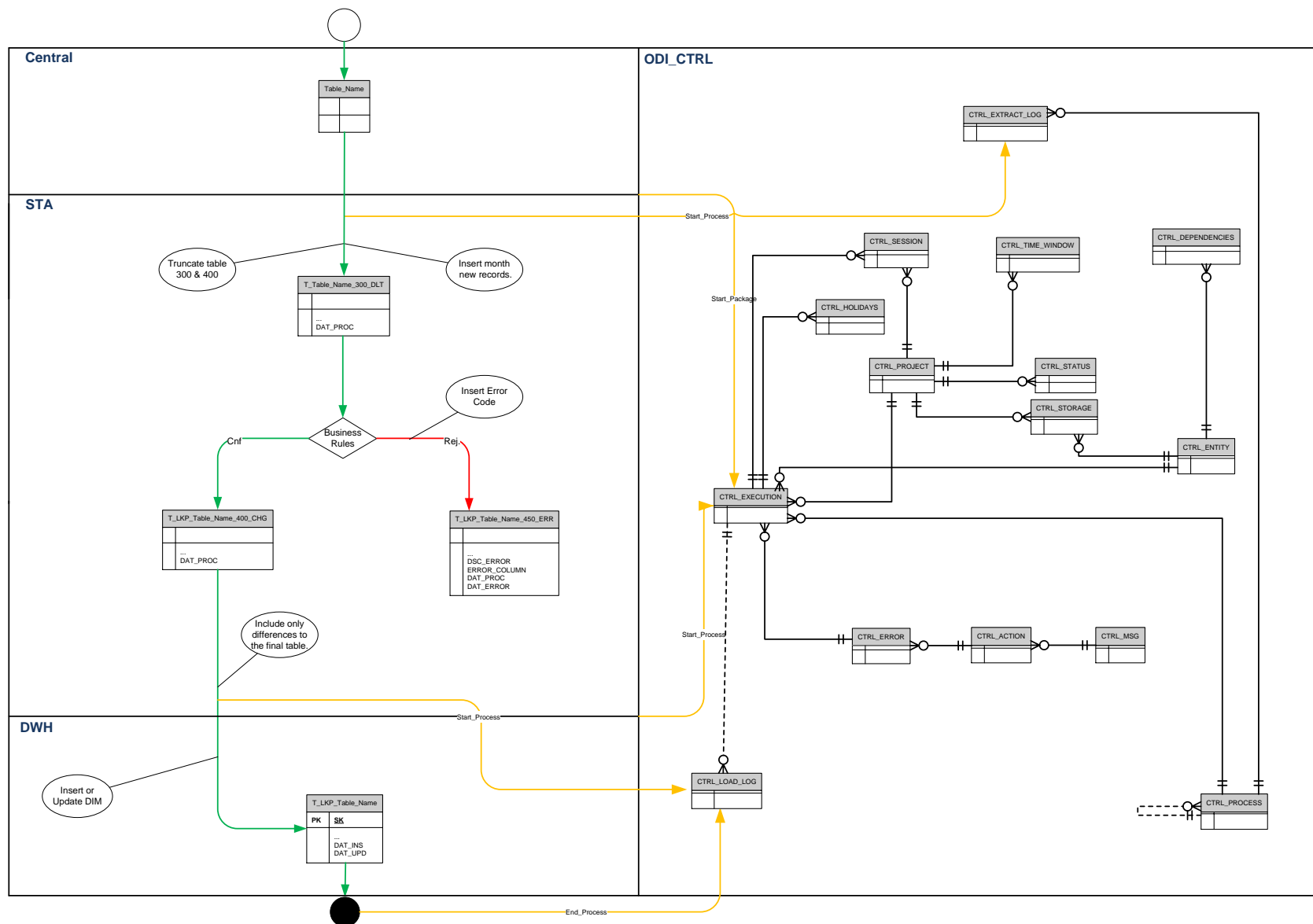


Figura 15 - Framework Carregamento de Dimensões

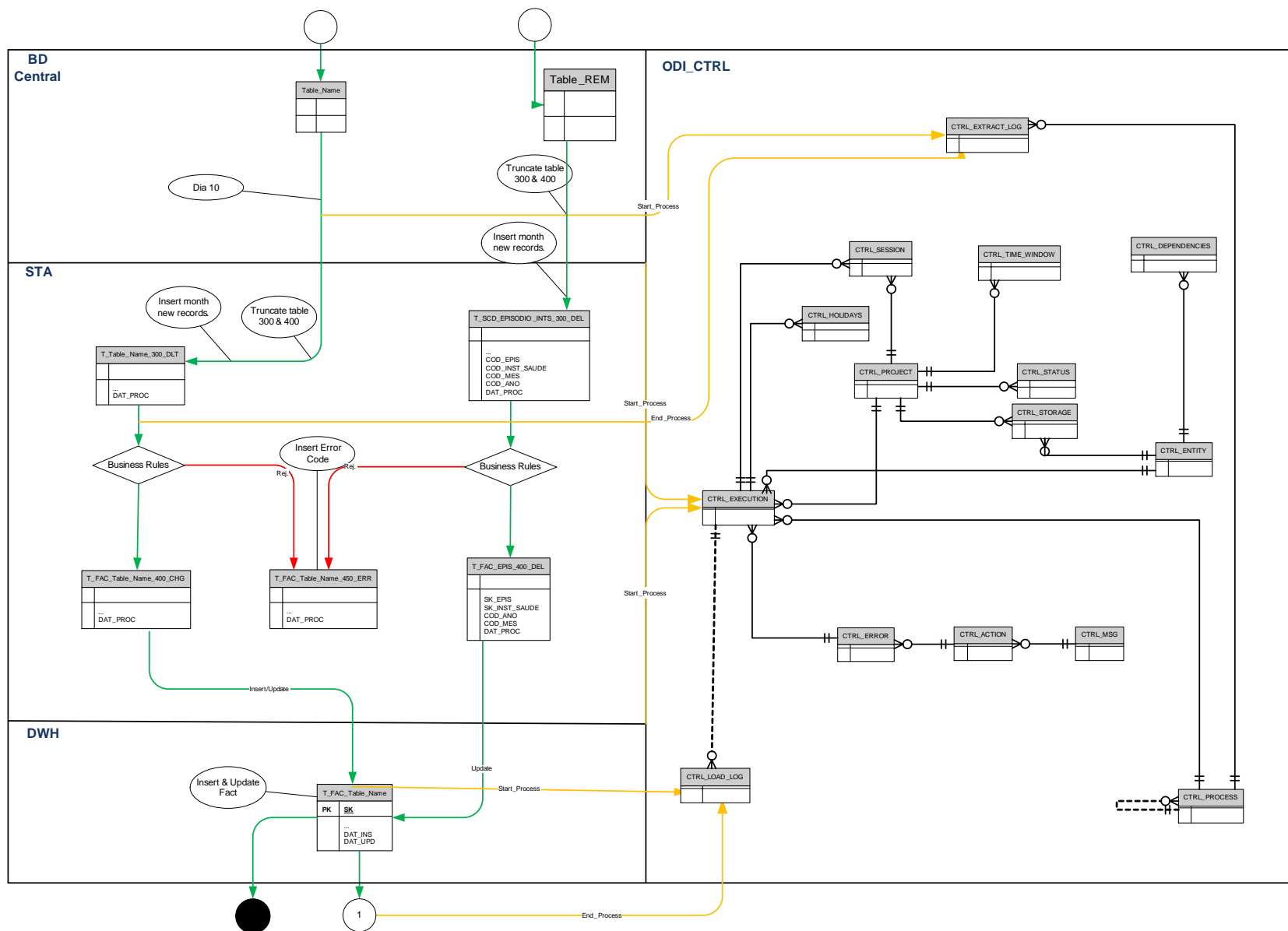


Figura 16 - Framework Carregamento de Factualis

- Logical Framework – Esta consiste numa camada de *views* lógicas definidas paralelamente com a camada de *reporting* num *schema* separado do DW, embora na mesma RDBMS física. Esta camada de *views* lê a informação directamente do DW, e constitui o que chamamos de modelo lógico, tratando-se ainda da fonte da ferramenta analítica OBIEE. Esta camada permite estruturar a informação por forma a dar resposta o mais directa possível aos requisitos de negócio, que é o objectivo dos *Keys Users*. Desta forma fica facilitada a modelação dentro da ferramenta analítica, OBIEE. Conseguimos ainda diminuir o tempo para efectuar a modelação no OBIEE. A razão para esta opção centra-se no facto do OBIEE em termos de modelação ser totalmente fiel ao *Star Schema*, e por essa razão é necessário simplificar o modelo para evitar complicações na tarefa de relacionar os dados. Existem regras a cumprir na criação desta camada de *views* que constituem mais-valias na sua manutenção e principalmente na *performance*.
 - Assim, não existem acessos directos a tabelas no DW, são criados objectos do tipo *Synonym*⁸ que acedem às mesmas. A vantagem desta situação prende-se com o facto de ser mais fácil quando necessário, apontar o DW para outra BD ou Schema. Para o efectuar bastará recriar os sinónimos, ficando as *views* a apontar para o novo *schema*, evitando assim termos de recriar o código de todas as *views* existentes.
 - Não existem *views* que utilizem outras *views* como fonte, o acesso por parte das *views* é sempre directo ao DW.
 - Todas as tabelas do DW são indexadas de acordo com o código gerado para estas *views*, criando índices de tipos específicos para dar resposta com a maior celeridade possível. Criando índices de tipo *Bitmap*, nas dimensões com baixo nível de granularidade, índices de tipo Local quando se tratam de tabelas particionadas e índices de tipo normal ou *B-tree* nas restantes situações. Os índices *Bitmap* só devem ser utilizados nas condições referidas, pois pelo facto de guardarem juntamente com a chave os *rowid*⁹ de cada registo tornam-se demasiado pesados noutras condições. Quanto ao funcionamento do índice local, este só pode ser utilizado nas tabelas particionadas. O facto de uma tabela particionada ser uma tabela lógica que é composta de N tabelas físicas, permite criar um índice local por partição e neste tipo de tabelas quando utiliza a indexação, parte do índice global da tabela que é composto pela chave da partição, recorre na partição pesquisada ao índice local que melhor se adequar à pesquisa desejada. Utilizamos ainda que de forma esporádica e apenas no carregamento de tabelas agregadoras, os índices baseados em funções, os *Function-based indexes*.
 - Na criação das *queries* utilizadas nas *views*, validando a sua *performance* no *explain plan table* do Oracle, justificamos a utilização de *Hint Rules*. Existem várias *Hints* que ajudam o motor de BD a tomar melhores decisões na construção do caminho que a *querie* vai percorrer na RDBMS e assim baixar o custo da mesma. Quando o tempo de resposta de uma *querie* é demasiado elevado, normalmente deve-se a uma má escolha na forma de relacionar a informação por parte do motor da mesma. Quando relacionamos, numa *querie*, tabelas que estão em utilizadores físicos de BD distintos e invocadas através de DB Link ou Sinónimos que referenciam um DB-Link, a RDBMS Oracle não tira partido da utilização dos índices. Esta é uma das situações em que devemos forçar a utilização desses índices. Outra situação comum e que justifica a utilização de *Hints*, será quando um universo gerados numa *querie*, funciona como uma *subquerie* e vai servir de fonte à principal, num SGBD Oracle, o que irá acontecer é que, por cada linha executada da *querie* principal, todos os universos dos quais esta depende, vão ser regenerados. De forma a acelerar a obtenção de resultados, podemos forçar a resolução deste subconjunto numa só iteração, ficando este em memória e utilizando este resultado para a *querie* da qual é dependente.

⁸ Nome alternativo para um objecto de Base de Dados

⁹ Pseudocolumn que contém o endereço de cada linha na BD.



- OBIEE – É a camada de mais alto nível e onde foi construído o modelo OLAP e o *reporting*, incluindo ainda a definição dos indicadores, métricas e respectivos factos.

Assim, com base no modelo do DW, construímos todas as hierarquias que havíamos definido ao nível do seu modelo base. Foi com base nestas hierarquias que se tornou possível dentro da ferramenta analítica fazer o *Drill Down* dentro dos factos, ou seja partirmos de um nível de agregação máximo associado a um facto e podermos descer até ao seu nível de granularidade mais baixo. Através desta descrição é possível perceber a vantagem da modelação do DW ser em *Snowflake*. Se tivermos uma estrutura normalizada, conseguimos percorrer as hierarquias como se tratassem de ramos de uma árvore. A versão do OBIEE utilizada no projecto tinha como limitação, o facto de não ser possível criar hierarquias multidimensionais, ou seja, onde exista mais de um caminho a seguir. Esta solução actualmente foi solucionada na versão mais actual, a 11G.

A título de exemplo do funcionamento das hierarquias, refiro a hierarquia de “tempo”, que é construída em outros projectos semelhantes, e inclui as dimensões Ano, Semestre, Trimestre e Mês, no seu nível mais baixo.

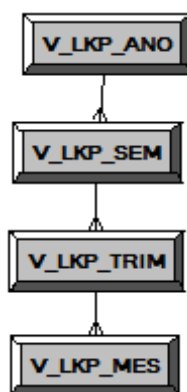


Figura 17 - Hierarquia de Tempo

Em termos da forma como a informação era apresentada, criámos 3 *dashboards* com filtros e *prompts* associados, ou ao nível do próprio *dashboard*, ou, de um separador do mesmo.

- BI Publisher – O OBIEE não permite a geração de documentação no formato MS Word de forma directa o que abriu espaço para uma nova necessidade após fechada a solução. Sendo esta nova fase, posterior à data determinada para fecho do projecto e, por questões inerentes a necessidades de outros projectos da BIMaven, fiquei sozinho no desenvolvimento desta componente.

Esta ferramenta tem a capacidade de interagir directamente com as *sources* do OBIEE, ou seja os relatórios neles criados. Assim, é possível importá-los directamente, após uma ligação ao seu servidor, para um *template* no formato RTF. Estes *templates* eram formatados em Linguagem XML, embora a maioria dos objectos fossem disponibilizados através de um *plugin* para o MS Word integrado no pacote do BI Publisher.

Foi necessário muito trabalho de “customização” dos relatórios pois o formato pretendido, para ser efectuado em XML, obrigava a muitas alterações às bases fornecidas, e à própria segmentação dos dados. Esta situação deve-se ao facto dos relatórios importados do OBIEE apenas conterem os campos, ou seja, os dados que pretendemos introduzir no BI Publisher, na realidade estávamos a importar as *queries* que este efectuava a BD, neste caso, na *Logical Framework*. Toda a apresentação e formatos foram trabalhados na camada do BI Publisher.

2.2.4 Análise de Resultados

A nível da modelação feita no OBIEE criámos um total de 600 indicadores, sendo entre 15 a 20 por cento a base para os restantes. Criámos 250 relatórios para dar suporte aos 3 *dashboards* e 2



documentos sob o formato de *templates*. Mencionando aquilo que foi imediatamente palpável na solução desenvolvida, o facto de um documento anual, que antes deste projecto, era produzido em 5 FTE, tenha passado a ser produzido em 30 minutos, é um indicador claro da valia do projecto.



3. Analise aprofundada do percurso profissional

3.1 Iniciação - NB

Com a entrada na Novabase, fui integrado na unidade de Business Intelligence e assumi a função de *Assistant Consultant*.

3.1.1 PT-Sistemas de Informação

O primeiro projecto para um cliente da Novabase foi numa grande companhia de telecomunicações a nível nacional, a PT, através da sua empresa de consultoria, a PT-SI [27], no caso foi um projecto da PT-COM [28] mas que envolvia todas as empresas do grupo. Foi um projecto de DQ onde utilizei como ferramenta, o Integrity [29], na altura propriedade de uma *SW House*¹⁰ Suíça, a Ascential. O produto era dos mais evoluídos a este nível e permitia ainda fazer integração a um nível muito básico. Em termos do *Data Profilling* possuía uma linguagem própria algo semelhante ao *Assembler* e que através da manipulação de ponderadores permitia na identificação das *strings* de *input*, a sua manipulação ou marcação de registos. Estas tarefas utilizavam algoritmos próprios da ferramenta, mas que, funcionalmente eram definidos a partir de regras de negócio criadas em sede de projecto.

Este projecto foi diferente dos que me surgiram posteriormente como consultor. Não só por ser de DQ, mas também porque era totalmente ligado a sistemas transaccionais, sem nenhuma componente analítica ou mesmo de modelação. Foi um mega projecto envolvendo inúmeros fornecedores desenvolvido de forma modular. Foi afirmado na altura como sendo o maior projecto de implementação TI a nível nacional [30].

A fonte que dizia respeito ao meu envolvimento era um ERP SAP R/3 onde estava toda a informação associada aos três grandes grupos de informação que eram a base do projecto, Clientes, Fornecedores e Materiais. Todos estes universos foram alvo de processos de Análise, Normalização, Consolidação e “Desduplicação”.

O processo era preparado de forma prévia ao longo de semanas e normalmente as execuções eram planeadas para ocorrerem aos fins-de-semana. Esta era uma forma de minimizar o risco, isto porque após os nossos processamentos tínhamos uma equipa de validações que estava pronta para analisar os dados por nós gerados. Em caso de falha ainda tínhamos tempo para fazer algum acerto ao nível do nosso processo e reprocessar. Era feita uma extracção no final do dia de sexta-feira e só após, feito o fecho aplicacional. Posteriormente era iniciado o processo, primeiro com uma extracção da informação utilizando uma rotina feita em SAP [31] que integrava algumas classes de linguagem C e depois desta execução, havia uma janela de horas para efectuarmos as execuções associadas aos nossos processos de DQ.

Após a extracção para ficheiros *plain text*, eram utilizadas algumas rotinas em HP-UX como o GAWK¹¹ que preparavam os dados para entrar nos processos do *Integrity*. Através destes

¹⁰ Empresa de desenvolvimento à medida de produtos de Software

¹¹ Biblioteca de funções estendida do AWK e com maior capacidade por exemplo no suporte de um maior número de colunas



processos eram seleccionados os campos pretendidos para o processamento de DQ, e posteriormente, eram invocados estes mesmos processos sobre a informação proveniente dos sistemas transaccionais. No final deste processo era feito um processo de *backup* de todos os ficheiros fonte e processados antes destes passarem por uma nova *interface* que os preparava para serem disponibilizados de novo e carregados no ERP SAP R/3.

Em termos de *timings* a minha passagem por este projecto durou 4 meses, tendo estado sempre incluído na equipa do cliente. Três meses com outro colega e no último mês sozinho.

3.1.2 ESI – Espirito Santo Innovation

A restante experiência com clientes da Novabase centrou-se na área da Banca onde estive em algumas áreas verticais dentro do grupo BES, um grande Grupo Bancário a nível Nacional e já com alguma presença a nível Internacional. A maior parte do tempo na sua empresa de GSI, ESI [33]. O projecto com maior destaque aconteceu na área da Recuperação de Crédito, sendo ainda hoje uma das fontes analíticas de referência.

Em termos temporais, a minha passagem por este cliente durou aproximadamente 2 anos.

O primeiro projecto foi o de maior relevância tendo-se tratado de um projecto *Turn Key*¹² que teve um tempo de implementação de aproximadamente 1 ano, e que após o *Roll Out* manteve um papel activo junto do mesmo, tanto na sua manutenção, como posteriormente, e por já estar adjudicado a outros projectos e áreas, na definição de novos requisitos junto dos *Key Users*. Isto tanto no planeamento de trabalho para a nova equipa de manutenção, como na definição de novas funcionalidades.

Em termos de definição, este projecto consistiu na criação de um modelo analítico, um DM, onde era incluída toda a informação associada aos Contratos de Créditos em situação de recuperação geridos por esta instituição. Diariamente era enviada pelos balcões, toda a informação, sob a forma de eventos, relativa a contratos activos e vencidos, desde as suas posições, movimentos, o *status*, e toda a informação processual associada.

A informação era integrada diariamente e existia *reporting* diário em MSTR 7i. A fonte do MSTR era a camada da *Logical Framework* criada na RDBMS Oracle.

Após a boa conclusão deste projecto, e dado o impacto positivo que o resultado deste causou, foi-me atribuída uma nota máxima na avaliação anual e atingi um novo nível de carreira onde assumi as funções de *Consultant*.

A partir da entrega do projecto anterior ao cliente fui convidado para integrar a equipa de DW do cliente e, até à minha saída da Novabase, não mais abandonei esta equipa.

Existiram vários projectos dentro desta entidade, possuindo esta, na altura, um dos maiores DW a nível nacional. Todos os processos introduzidos e integrados nos processamentos quer diários, mensais ou anuais obedeciam a uma *framework* específica, e que, era uma garantia de estabilidade e fiabilidade para quem estava incumbido da manutenção destes sistemas.

Os processos de integração no DW tinham obrigatoriamente que passar por três utilizadores de BD, o ODS, a STA e o DW.

Toda a informação era enviada em ficheiros provenientes de várias áreas e aplicativos, incluindo a informação proveniente do AS400 introduzida nos balcões. Esta informação era integrada no ODS normalmente através de processos Datastage [32] ou *Shell Script* Unix. Independentemente

¹² Projecto chave na mão ou fechado



da máquina em que estes fossem executados, existia uma ferramenta que centralizava e calendarizava todos os processos coordenando a ordem e tempo de execução dos mesmos, o Control-M¹³. Esta ferramenta invocava scripts Unix HP-UX onde na sua chamada podiam ser incluídos algum tipo de parâmetros fornecidos por variáveis da própria aplicação. Toda esta metodologia, exigia o preenchimento de *templates* específicos de modo a manter documentação actualizada de todas as interações a que obrigava cada um dos novos processos inclusos. Nestes estava toda a informação associada ao processo, tal como todas as suas dependências ou variáveis invocadas.

Quanto a *framework*, esta foi desenhada internamente e consistia numa série de metodologias de apoio aos processos de carregamento do DW em PL/SQL, e em especial no ODS. Existiam funcionalidades pré-definidas como funções e parâmetros genéricos, que eram utilizadas em todos os processos no *logging* ou nas suas operações.

Toda a estrutura da *framework* era totalmente “customizável” e adaptável, e permitia, manipulando apenas tabelas de parâmetros, e os parâmetros de chamada dos *Packages*, *Functions* e *Procedures* alterar as fontes e destinos de todos os processos, quer em termos de Schemas ou BD.

A *framework* tinha como principais objectivos:

- Melhorar a *performance* dos processos na STA:
 - Reutilização de módulos processuais evitando trabalho repetitivo;
 - Uniformizar as ferramentas de desenvolvimento;
- Facilitar a manutenção da STA:
 - Optimizando a quantidade de informação processual;
 - Garantindo processamentos parciais;
- Melhorar a adaptação às necessidades do negócio:
 - Maior flexibilidade para inclusão de novos requisitos;
 - Baixar os tempos de desenvolvimento;
 - Melhorias na detecção e correcção de informação incorrecta;
- Adaptação a diferentes sistemas.

Uma componente muito interessante em termos de controlo do risco associado aos processamentos estava relacionada com o processamento das dimensões. Embora este fosse diário, existiam 2 utilizadores de BD associados, um que continha a imagem do dia processado e outro tinha sempre o dia anterior. Para garantir o bom funcionamento de toda a camada de *reporting* estava criado um sinónimo ao nível da BD que servia como base para todas as *queries* efectuadas pelo modelo do MSTR. Assim, a grande vantagem residia no facto de em caso de irregularidades no processo havia uma forma fácil de ter toda a camada de *reporting* funcional rapidamente, bastando “reapontar” o sinónimo existente para o utilizador com os dados do dia anterior. Esta solução de *recovery* não seria uma solução totalmente satisfatória mas garantia o bom funcionamento do sistema e era facilmente implementável.

Em termos da minha participação nos projectos seguintes na ESI, o foco foi a área de *Ratings* e *Scores* de clientes. Devido à iminente entrada em vigor do novo acordo “Basileia 2”, iniciei o projecto de criação da sala de mercados e participei ainda na integração de novas entidades estrangeiras no DW, como a rede de balcões de Espanha, Inglaterra e outros. Toda esta integração tinha que ser feita nos mesmos moldes das entidades réplicas que já existiam no DW.

Fiquei ainda durante a minha passagem, responsável por um protótipo que consistia na geração de um DM que antecipava as análises que se sabiam obrigatórias no futuro “acordo Basileia 2”. Era um protótipo com análises mensais em que não existia qualquer preocupação em termos de *performance*, quer a nível do seu tempo de processamento, quer depois na análise da informação

¹³ Ferramenta da BMC para calendarização de Batch



gerada. A posterior análise desta informação era feita *Ad-Hoc* através de *querying* SQL, ou no Excel, a partir de extracções de informação.

3.1.3 ESCOB – Espírito Santo Cobranças

Tive uma intervenção pontual, com a duração de um mês, numa das fontes do projecto que desenvolvi na ESI referente à recuperação de crédito. Tive que intervir neste processo operacional na área de cobranças. Na fonte SIG, que consistia numa RDBMS em SQL Server 2000 que tinha uma componente analítica ROLAP em Microstrategy directamente sobre esta última. Esta intervenção devido a questões de segurança da própria instituição teve que ser efectuada directamente no servidor de integração, um *rack* com Intel Xeon. Esta intervenção consistiu na adição de uma nova fonte de dados que passou a fazer parte desse SIG tal como a inclusão de *reporting* sobre esta nova informação carregada.

3.2 Evolução - Staples

A integração na Staples Office Centre [34] permitiu-me assumir diferentes funções e explorar com maior profundidade uma área em que eu tinha menor experiencia, os sistemas transaccionais.

Nesta vertical, todas as interfaces aplicacionais existentes tinham sido criadas em *Shell Script* Linux Suse¹⁴ invocando Oracle SQL Plus¹⁵ ou em Pro*C¹⁶ compiladas e também invocadas via Linux Suse. Estas interfaces eram invocadas de 4 formas possíveis, manualmente em casos de recovery de alguma situação não esperada, de forma automática via Shell script no *Day Run* e no *Night Run*, ou ainda, via aplicacional, através de *Forms* ou VB.

Para além destas interfaces existiam algumas rotinas criadas para a impressão de etiquetas e listagens, sendo que as primeiras eram criadas através de scripts que invocavam código ZPL, no caso das Sinalética e Rail Card¹⁷, ou então, utilizando também o Pro*C nas de Vitrina. Já as listagens eram habitualmente geradas em *Shell Script* invocando SQL Plus, sendo estas redireccionadas directamente para as impressoras de cada loja.

Em termos aplicacionais tínhamos como *front-end*, um ERP que era totalmente “customizado” em tecnologias Oracle. A RDBMS era uma Oracle 9i, posteriormente migrada para 10G. Sendo toda parte visual programada em Forms e Reports 6i. Possuíamos também o Oracle Finance¹⁸ e o OTS integrado no E-Business Suite¹⁹.

A nível de *reporting* analítico *ad-hoc* possuíamos uma aplicação criada *in-house* em VB que invocava os *Shell Scripts* com SQL Plus. Como ferramenta analítica com alguma complexidade possuíamos o Oracle Discoverer 4i, mas, era apenas utilizado em projectos ligados a área financeira e nunca vingou em termos de referencia interna.

Relativamente à aplicação de *reporting* desenvolvida *in-house*, que já referi, esta possuía um *front-end* desenvolvido em VB onde através de caixas e selectores específicos nos permitia executar *reports* com parâmetros pré-definidos e gerá-los directamente para um output definido *on-time*, como seja o ecrã, ficheiro, impressora ou directamente para um *mail*. Para criar um *report* era necessário gerar um script em *Shell Script* Unix, que quando executado gerava um ficheiro de output do tipo *text file* sendo este depois redireccionado pela aplicação para um destes outputs específicos.

¹⁴ Versão do SO Linux original da Novell

¹⁵ Linha de comandos Oracle que permite executar PL/SQL e SQL

¹⁶ Linguagem de Programação C com SQL Oracle e Sybase embebida

¹⁷ Impressora manual de etiquetas

¹⁸ ERP integrado no E-Business Suite com foco na área financeira

¹⁹ Suite Middleware da Oracle que inclui produtos de ERP, CRM e SCM



Na “customização” do ERP participei num projecto interessante de criação de novos ecrãs para o SPV.

Este projecto envolveu a utilização das tecnologias Oracle Forms e Reports 10. Foi um projecto importante pois até aqui e para além da manutenção de todo o ERP, os Forms que havia criado de raiz tinham uma baixa complexidade. Este conjunto de ecrãs para o SPV permitiu-me evoluir bastante no conhecimento desta tecnologia, assim como na *interface* com passagem de parâmetros entre o Forms e o Linux.

Este projecto teve como componentes mais interessantes não só a reformulação de todos os ecrãs que já existiam, onde introduzi por exemplo os menus com *drop downs* para facilitar a navegação, mas também, a centralização de todo o sistema, e por conseguinte, das suas *interfaces*, no armazém em vez de como anteriormente deslocalizado em cada loja.

Assim foi criada uma nova *interface* com a aplicação WPMS, o que obrigou a trabalho com o fornecedor externo desta ferramenta. Foi feita ainda uma preparação embora não tendo sido concretizada na altura para no futuro permitir uma *interface* directa com os reparadores. Neste módulo existia um total de 8 ecrãs que interagiam e permitiam a navegação entre si, sendo em cada um gerada uma instrução para impressão de vários tipos possíveis de guias de reparação. Estas eram impressas através da execução directa de *Shell Scripts* e onde seria passado como parâmetro a escolha da impressora, para a qual deveria ser enviada. Os nomes das impressoras eram partilhadas de loja para loja, e assim sendo, a partir do momento em que efectuássemos o *login* numa determinada loja, a impressão era redireccionada para o IP correspondente à impressora esperada.

Em termos do projecto com maior relevância em que participei, terá sido a centralização do ERP. Tratou-se de um projecto de elevada complexidade não só pelas componentes que foram contempladas, como também pelo facto de ser um projecto estratégico para a companhia, o que acrescentou uma pressão adicional à equipa.

O projecto teve 4 componentes principais:

- Solução de Hardware
- Centralização do RDBMS
- *Front-End* ou componente aplicacional
- Programas ou Rotinas

Em termos de solução de Hardware existiram várias preocupações como sejam incremento de *performance*, estabilidade, fiabilidade e o TCO. Foram consultados dois fornecedores de HW e a escolha da solução acabou por residir na diferença no TCO, pois ambas as soluções eram muito semelhantes nos restantes aspectos.

Em termos específicos a solução construída era uma arquitectura transaccional de alto débito constituída por um *rack* ²⁰ com *load balance* ²¹ ao nível do SGBD gerido através do Oracle RAC. Esta arquitectura era escalável e com grande capacidade evolutiva.

Ao nível do servidor aplicacional existiam duas máquinas embora ambas partilhassem o mesmo storage. O *storage* era *iSCSI* com interface *Fibre Channel*.

Tratou-se de um projecto complexo devido ao facto do ERP estar presente em 26 lojas num servidor central e ainda fazer o mesmo ao nível da RDBMS, onde existiam 26 instalações físicas. Para além disso, fizemos uma migração de versões de BD, neste caso de Oracle 9i para 10G. Para simplificar este processo de centralização e devido às inúmeras dependências existentes foram criados utilizadores lógicos de nível global que traduziam os utilizadores físicos já

²⁰ Plataforma escalável destinada a infra-estrutura de IT organizacional.

²¹ Repartição de carga de um serviço por mais de um recurso da infra-estrutura.



existentes. A vantagem seria no facto de permitir manter a estrutura de *logins* e utilizadores exactamente igual à que estava ao nível aplicacional. No momento em que um *login* era efectuado, e de forma transparente para o ERP, utilizando as chaves FGAC através de uma *Policy* convertia o *login* aplicacional, lógico, num *login* físico. Com isto conseguia-se aceder a aplicação e esta à BD. Esta solução minimizou em grande parte o risco associado ao projecto pois garantia a manutenção da estabilidade já existente nos módulos utilizados.

3.3 Consolidação - BIMaven

Na integração na BIMaven Consulting tive como foco o meu regresso ao BI, e hoje, estou convicto que fiz uma boa opção uma vez que desde a minha entrada tenho sido exposto a desafios diversos e interessantes, que me proporcionam uma aprendizagem constante. Trata-se de algo altamente motivante e que faz com que ainda hoje mantenha uma alegria quando me encaminho diariamente para o meu trabalho.

Logo após a minha entrada estive envolvido num projecto no grupo **Media Capital** [35] onde fui desenvolver uma *interface* em Integration Services 2005, de forma a integrar informação proveniente de SAP BW num modelo ROLAP, Microstrategy 8i.

Após este projecto comecei pela primeira vez a entrar no negócio da saúde. Na **ACSS** [36] estive inicialmente envolvido no desenvolvimento de várias propostas comerciais inclusive para resposta a concursos públicos. Mais tarde estive envolvido num projecto ligado à área de Financiamento que permitiu a geração de *reporting* mensal. Neste projecto foi efectuado um trabalho de reorganização das necessidades da unidade em questão, tendo sido introduzidas várias metodologias de trabalho ao nível da documentação tipo que permitem hoje um controlo do formato e estrutura de toda a informação gerada e disponibilizada.

Participei na resposta a um RFI para o **MDN** [37] com o objectivo de criar um sistema analítico a partir de uma fonte SAP BW.

Estive depois envolvido em dois projectos na Indústria através da **Efacec** [38]. Um primeiro, apenas de modelação ao nível da camada de *views*, a *Logical Framework*, que iriam alimentar uma camada de *reporting*, e, um segundo, mais extenso onde foram desenvolvidas interfaces de carregamento de todo um modelo já existente e que estavam desenvolvidas anteriormente numa tecnológica já obsoleta.

Participei depois numa experiência totalmente nova para mim no mundo laboral. Fui orador numa palestra no CCB no âmbito de um evento da **Oracle**, com a qual, a BIMaven mantém uma parceria. O tema apresentado foi “Consolidate to Grow”.

Posteriormente, voltei à área do retalho, onde estive na **FNAC** [39]. Foi um projecto em que estive envolvido apenas na fase de análise e, o principal objectivo foi também tentar captar necessidades que existissem dentro da FNAC, de forma a estruturar futuramente uma oferta que se adequasse.

Estive depois envolvido num projecto na área dos seguros na **Groupama Seguros** [40], foi um projecto numa área inovadora para mim, tendo participado na proposta e em toda a fase de análise.

Voltei a área das TELCOS onde estive na **ZON** [41], envolvido em dois projectos, ambos tinham como objectivo a construção de modelos analíticos ROLAP.

Actualmente, estou a desenhar uma solução BI Real-Time na área da comunicação social envolvendo algumas componentes da oferta de *Middleware* da Oracle. As tecnologias utilizadas são parte integrante da plataforma de Fusion Middleware da Oracle, o SOA Suite e ainda o ODI.

Assim, vou agora fazer uma referência mais específica a cada um dos clientes e aos respectivos projectos por onde passei e respectivas particularidades e adversidades de cada um deles.



3.3.1 BIMaven

Dentro da BIMaven e falando de projectos internos assumi dois tipos de papel, essencialmente, na análise de sistemas por forma a poder ou não gerar possíveis propostas, na investigação de novas tecnologias e áreas de mercado.

Foi no primeiro papel que estive envolvido na criação de duas propostas para a ACSS.

A primeira tratou-se da resposta a um concurso público, ao qual após uma análise extensa e de algum trabalho preparatório na proposta, se chegou à conclusão que a relação entre o ROI e o risco, não era positiva a favor do primeiro. A solução requerida pelo caderno de encargos consistia na criação de um sistema aplicacional e analítico onde fosse possível manter a informação sobre todas as entidades do SNS, todas as suas características e existências, assim como introduzir nova informação sobre as já existentes, ou introduzir novas entidades. Para esta ultima funcionalidade seria necessário criar um sistema aplicacional de raiz, “customizado” à medida, situação esta que estava fora do *core base* da BIMaven. Por essa razão decidimos avançar com uma parceria que sustentasse essa capacidade para efectivamente assumirmos o projecto. Toda a componente analítica foi então definida por nós, tal como a respectiva análise.

Nesta situação, embora o resultado final não fosse o pretendido, a tarefa acabou por ser aliciante e conter muitas novidades para mim. A resposta a um concurso público reveste-se de particularidades relacionadas com a forma como se estrutura a proposta para ganhar. Esta tem que se focar muito para além da qualidade da solução, a pontuação é o mais importante e tentar obter a máxima em todos os critérios, resultará numa eventual adjudicação. É necessário muitas vezes sacrificar a qualidade da solução e, principalmente da equipa de projecto, por forma a conseguir-se um equilíbrio que motive a resposta à mesma. Eventualmente quando os critérios são demasiado rígidos e apertados, ao ponto de o risco da solução final poder vir a não ter a qualidade que um implementador gosta de se orgulhar e mostrar ao mercado, pode conduzir por vezes à situação de desistência, mesmo após toda a proposta ter sido criada. Isto porque por vezes numa primeira visão sobre o caderno de encargos, e mesmo com gestores de projecto experientes, torna-se difícil aquilatar de imediato se o projecto pode ter um interesse real para o implementador. Só em companhias de grande dimensão, com equipas muito estabilizadas e com SLAs já muito bem definidos, se pode rapidamente analisar o real interesse na resposta a um concurso. E ainda assim, esta situação não é garantida pois inúmeras vezes só após uma análise exaustiva de todos os parágrafos de um caderno de encargos, se consegue atingir o real esforço necessário para levar a cabo a solução.

A segunda proposta para a ACSS que efectuei teve como resultado uma adjudicação culminando assim na sua posterior implementação. Aqui assumi um papel de *Team Leader end-to-end*, ou seja desde a criação da proposta comercial, a sua entrega, todo o desenvolvimento e gestão da equipa de projecto até à sua passagem ao ambiente produtivo. O objectivo destes sistemas era a substituição das interfaces em PL/SQL já existentes, e a sua substituição por uma ferramenta de integração. O segundo produto da solução foi a criação de um sistema analítico tendo por base as BD fonte das interfaces já referidas.

Assim, escolhemos como ferramenta de integração o ODI, devido a inúmeras vantagens que apresenta não só em termos do paradigma relativamente à concorrência, como também em termos de conectividade. Já para a criação deste sistema analítico, propusemos a criação de um sistema de DW utilizando uma RDBMS Oracle 11G para o qual tivemos que definir toda as estruturas e modelação, assim como as interfaces das mesmas. Como ferramenta de integração mantemos a aposta no ODI 10G e, como ferramenta de *Reporting*, o OBIEE 10G. Tratou-se de uma proposta bastante extensa e complexa, e que, no final teve uma excelente aceitação. Nesta foram invocadas todas as metodologias da BIMaven, na gestão de projecto, gestão de processos, testes, auditoria e riscos.



Após estas duas propostas surgiu mais tarde a hipótese de fazer uma resposta a um RFD para um projecto de CRM/DQ para a Groupama Seguros. A proposta teria que ser criada em parceria com outra consultora pois as nossas competências não cobriam todo o desenvolvimento que teria que ser assumido. Assumi a liderança da equipa que faria a proposta e que foi desenhada em pouco tempo pois só havia dois dias para a entregarmos. Ainda assim dada a qualidade da mesma e após uma reunião onde fomos defender a mesma acabamos por ganhar e fomos escolhidos para implementar o projecto.

3.3.2 Media Capital

O objectivo aqui era a construção de um modelo analítico em Microstrategy 8i, sendo as fontes em SAP BW. Embora o Microstrategy tivesse um conector para SAP BW, que permitiu construir todos os modelos, existia alguma informação financeira das rubricas de balanço que não era possível integrar.

Assim, foi necessário construir um processo alternativo que alimentasse o Microstrategy. O repositório destino escolhido para o processo foi a RDBMS MS SQL Server 2005. A razão para tal escolha está muito relacionada com as políticas comerciais das empresas fornecedoras. Neste caso, falando da MS, existem muitas empresas que adquirem licenciamentos corporativos com condições vantajosas e, inúmeras vezes, são disponibilizados produtos sem custos, isto é, se a ferramenta não for utilizada não há qualquer custo, caso contrário, este custo é adicionado ao licenciamento corporativo na renegociação do contrato, habitualmente com condições muito favoráveis relativamente a outros fornecedores. No caso da MS, esta tarefa é habitualmente facilitada em grandes organizações, pois a maioria delas utiliza o MS Office para todos os seus colaboradores de áreas funcionais e de investigação e assim sendo, fica facilitada a tarefa de conseguir vendas de produtos de áreas diferentes.

Foi assim disponibilizada uma VM Windows onde instalei o MS IS 2005. Defini os acessos à BD destino em MS SQL Server 2005 e pude assim desenvolver o novo processo, o qual já havia preparado previamente de forma a ser mais rápido a sua integração. Assim só serem necessárias fazer algumas adaptações.

Este trabalho prévio, mais do que apenas preparar o processo consistiu na aprendizagem desta tecnologia, o IS, isto porque embora já conhece as RDBMS SQL Server, só conhecia até então a versão 2000 e, nessa altura, esta ferramenta de integração não existia. Como não havia tempo para recorrer a formações, pedi ajuda a um colega da empresa que conhecia a ferramenta pelo menos nas suas *tasks* mais básicas, e comecei a desenvolver o processo. Tal como em muitos projectos anteriores, resolvi recorrer à ajuda na Web do Google, para pesquisar informação que me ajudasse a solucionar os problemas que me iam surgindo. Fiquei surpreendido pela falta de informação a altura relativa a esta ferramenta de integração. Estando habituado a pesquisar informação para ferramentas Oracle, onde se encontra efectivamente uma grande quantidade de *posts*, verifiquei que não existe informação muito diversificada para o MS IS.

O IS é uma ferramenta um pouco enganadora numa primeira abordagem, isto porque parece muito simples de utilizar. Efectivamente é muito fácil construir uma *interface* simples e num ambiente que já se encontre estável, esta vai funcionar com toda a certeza, mas apenas no caso as manipulações a efectuar nos dados sejam básicas. Quando queremos explorar as *tasks* mais complexas, em que é necessário usar vários objectos para o conseguir, nem sempre somos bem-sucedidos de forma imediata.

Em termos de *performance* descobri, na altura que quando utilizamos PL/SQL *tasks* com implementação de código directo, que o SQL Server melhora a sua *performance* com *queries* em SQL 92, em detrimento do 86.



Ainda assim, e após o ambiente estabilizado, foi muito rápido adaptar o processo que havia sido feito por forma a responder ao esperado. Este processo consistia na leitura directa a ficheiros que eram extraídos do SAP BW num determinado dia do mês, sendo que o IS através do seu *Scheduler* nativo, na data esperada, arrancava com este processo.

O objectivo do processo foi carregar duas tabelas factuais com informação relativa às rubricas de balanço mensais das empresas do grupo Media Capital. O processo de ETL foi constituído por 7 ficheiros do tipo csv, cada um pertencente a uma empresa e com exactamente o mesmo formato, Ou seja, 6 campos com separador “;” e a primeira linha com o cabeçalho.

Após o arranque, a primeira *task* fazia um *ping* para verificar a existência no directório definido dos ficheiros esperados. Esta *task* ficava a correr em background e sempre à espera que o *ping* retornasse o “OK”. Depois os ficheiros eram copiados para outro directório definido como directório de trabalho e, só partir deste, iriam ser manipulados pelas restantes *tasks*, primeiro sendo copiados directamente para tabelas que constituíam a STA. Posteriormente, a partir destas, era feita a manipulação e o carregamento das tabelas finais, sendo as tabelas destino uma factual e duas dimensões.

Todo o fluxo de execução do processo era feito posteriormente no SQL Server Management Studio.

3.3.3 ACSS

Na ACSS estive envolvido em dois projectos, que decorreram simultaneamente, à excepção da data de arranque que estava desfasada em um mês. Foi um projecto *Turn Key* com a duração de 18 meses. Tratou-se de um processo extenso e muito interessante no desenvolvimento de uma solução *End-to-End*, desde a *interface* operacional até ao modelo analítico aplicacional.

Estes projectos merecem ser descritos com pormenor pois espelham o que a maioria dos consultores de BI esperam um dia encontrar na sua carreira e muitos não conseguem, a solução *End-to-End*. Até esta data eu conhecia todas as peças da engrenagem de um processo desta natureza, mas, nunca tinha estado envolvido desde o momento zero até a passagem a PRD, em projectos deste nível. De referir que estes dois projectos foram geridos por mim em simultâneo, embora tenham tido interlocutores diversos e âmbitos bem diferentes.

A primeira preocupação após a definição da data de arranque do mesmo foi a definição da equipa. Tendo em conta a natureza do processo, iriam ser necessários incluindo o *Project Manager*, 4 (quatro) elementos. Eu como *Project Manager* teria que assumir o desenho de toda a arquitectura a desenvolver, necessitava de dois elementos totalmente operacionais e mais um que me desse apoio em toda a fase de análise funcional e técnica e, por fim, no desenvolvimento da fase de *reporting*. Foi necessário recorrer à subcontratação de um dos elementos da equipa. Assim, contactei algumas empresas conhecidas na área de tecnologia como especializadas no chamado *Body-Shopping*. e comecei a receber currículos que se enquadrassem. Relativamente a este processo, houve algumas conclusões interessantes que retirei, e que considero importantes partilhar pois espelha um pouco o mercado ao nível do BI e Integração que existe em Portugal na actualidade. O perfil procurado seria alguém com as seguintes características:

- Conhecimento em PL/SQL Oracle e na criação de interfaces;
- Formação académica relevante;
- Experiência profissional superior a 1 ano em ambientes de DW;
- Conhecimentos de estruturas de Base de dados Oracle em ambiente distribuído;
- Conhecimento em ferramentas de integração visuais, sendo o ODI/Sunopsys preferencial.
- Conhecimentos consolidados de OBIEE;



- Facilidade de relacionamento interpessoal e trabalho em equipa;
- Capacidade de pensamento analítico;
- Fluência em Inglês.

Embora estejamos a falar da franja de mercado onde se situam as empresas de *Body-Shopping*, que como é sabido, possuem perfis muito diversificados e normalmente focados em várias áreas. O seu objectivo é dar resposta às inúmeras solicitações que lhes vão surgindo. A conclusão sobre este processo foi algo inesperada. O conhecimento em ferramentas de integração visuais constatou-se que era inexistente na maioria dos candidatos, e o mesmo em ambientes de DW. Alguns conheciam superficialmente por terem estado associados a aplicações que alimentavam ou que recebiam informação dos mesmos, mas poucos conheciam profundamente em que consistia um DW, exceptuando que era um repositório de dados. Conhecimento em ODI é coisa quase inexistente no mercado nacional. Esta última situação deixou-me surpreendido tendo em conta o facto de no passado ter estado integrado nestes ambientes, onde existiam equipas de várias consultoras com inúmeros elementos. A única coisa que havia efectivamente muitas respostas positivas era no conhecimento das BD Oracle em *Stand-Alone* ou distribuídos, e, no domínio da linguagem PL/SQL. Após várias entrevistas chegámos a um candidato que embora não conhecesse o ODI, tinha todos os restantes requisitos pretendidos. A entrevista decorreu com sucesso e assim acertámos a sua contratação nessa altura.

Após o processo de selecção da equipa, e ter vencido mais um desafio, marquei a reunião de *Kick Off* para dar início ao projecto. Na reunião apresentei os seguintes pontos:

- A metodologia de desenvolvimento com cada uma das fases, a sua descrição e os seus produtos;
- A metodologia de PM *on-field*, com o planeamento da periodicidade das reuniões;
- Identificação dos riscos associados, o tipo de impacto causado, o seu *status* e as acções que deveriam ser tomadas de forma a evitá-los ou controlá-los;
- Um macro plano temporal das várias fases do projecto;
- A definição da equipa com os respectivos papéis e responsabilidades;
- A Matriz de responsabilidades da BIMaven e da ACSS, onde estavam enquadrados todos os elementos envolvidos no projecto, desde o *Sponsor*, o PM, as equipas funcionais e técnicas.

Devido a serem dois projectos a decorrer em simultâneo tive que definir os tempos de arranque de ambos e, esta era uma situação que teria que ser considerada, pois havia estruturas comuns entre eles. As estruturas destino do projecto operacional eram a fonte do analítico. A divisão foi definida num projecto que teria como objectivo substituir as interfaces operacionais existentes em PL/SQL por processos desenvolvidos em ODI, e, outro de criação de um modelo analítico onde seria feita toda modelação para criação e “população” do DW, a criação de um modelo lógico e de negócio já dentro da ferramenta analítica. Esta divisão foi possível devido ao facto de todas as estruturas do modelo operacional se terem mantido praticamente inalteradas com este novo processo, e assim sendo, as estruturas base necessárias ao modelo de negócio estavam disponíveis e com dados disponíveis para análise.

Estes projectos estão descritos na sua componente técnica nos capítulos 2.1 e 2.2.



3.3.4 Groupama Seguros

A minha relação com a Groupama iniciou-se com a criação da proposta para um projecto de CRM/DQ. Foi a minha primeira participação num projecto totalmente focado no CRM, onde investiguei sobre as metodologias aplicadas.

O grande objectivo deste projecto seria a melhoria da qualidade da informação de Cliente. O segundo objectivo seria alterar o conceito de Cliente, visto que uma pessoa introduzida num sistema deste género pode não se tratar de um cliente mas de um *prospect*. O objectivo final do projecto seria melhorar a ferramenta de CRM já desenvolvida *In-House*, e após este processo, gerar campanhas de Marketing direccionadas. Para atingir esta situação precisávamos de completar e enriquecer a informação de cliente.

O problema da qualidade da informação, existe em quase todas as organizações e entre outras, tem a ver com a não existência de processos informáticos de DQ, o facto de muitas das regras existentes internamente não passarem de normas não aplicadas informaticamente ajuda a que estas situações se propaguem.

Dividimos o projecto em 6 etapas, que consistiam numa primeira fase de análise AS-IS, com o objectivo de obter uma visão global das características do seu Sistema de Gestão de Clientes e definir com exactidão o âmbito do projecto. Assim conheci todas as formas de entrada de clientes na BD Central da Groupama.

Na construção da especificação funcional propus melhorias ao nível das regras a implementar informaticamente e de forma centralizada. Esta centralização foi fundamental uma vez que se verificou existirem várias portas de entrada para a informação de cliente, e era fundamental filtrá-la antes de chegar às estruturas finais.

Com base nos resultados da fase anterior, foi desenvolvido um módulo aplicacional Web em que eram utilizadas as regras de DQ, que definimos e que tinham como objectivo fazer o *profiling* da informação, ou seja, detectar por exemplo clientes já existentes para que canalizasse o utilizador para um ecrã de actualização da sua informação pessoal, no lugar de permitir a introdução repetida desse cliente.

Teríamos ainda de solucionar o problema das duplicações de registos de clientes já existentes. Para ajudar nesta tarefa recorremos à ferramenta *Open-Source*, Talend Open Profiler [42], a partir da qual gerámos relatórios sobre a qualidade da informação e foi proposta uma nova hierarquia de registos de pessoas com o conceito *Master / Slave*. Foram definidas regras que garantiam a unicidade de cliente na BD e com base nestas e na manipulação de ponderadores na ferramenta de DQ, gerámos um universo de possíveis duplicados de clientes que foram apresentados à Groupama. De referir que a BD analisada era AS/400. A minha participação no projecto terminou com esta fase da análise AS-IS.

3.3.5 FNAC

O projecto na Fnac foi um regresso à área do Retalho, onde após a minha passagem pela Staples, existiam razões óbvias para o meu envolvimento neste projecto.

O objectivo deste projecto foi a criação de uma nova camada de *reporting* em Microstrategy sobre uma fonte de dados analítica Teradata. Trata-se de uma DBMS muito utilizada neste tipo de sector onde é necessário um motor de BD de alta disponibilidade. A DBMS Teradata disponibiliza módulos aplicacionais de BI, mas de baixa complexidade comparativamente ao MSTR. As características base desta BD são a alta capacidade de processamento em paralelo e a



escalabilidade devido a ter uma boa gestão do balanceamento com vários servidores disponíveis. A conectividade com esta no nosso caso foi via ODBC, embora também pudesse ser feita via JDBC ou com plugins específicos.

Voltei a participar apenas na fase de análise, pois entrei no projecto com uma perspectiva mais comercial. O objectivo consistia em validar os sistemas existentes e as perspectivas que poderiam existir na implementação de novas soluções. Não participei activamente no desenvolvimento da solução, apenas na especificação funcional.

3.3.1 EFACEC

Neste cliente da área da indústria, estive envolvido em dois projectos. Um primeiro, onde efectuei apenas um processo de modelação de informação, utilizando *queries* numa BD SQL Server 2005, e que, posteriormente, eram consumidas pelo Microstrategy 8i. Neste projecto assumi um papel puramente técnico. Esta camada tinha também como objectivo a tradução para Inglês de todos os objectos existentes na BD, que se deveu à internacionalização da empresa, que hoje já conquistou inúmeros mercados a nível mundial.

O segundo projecto foi complexo porque consistia na substituição do processo de carregamento diário de um DW tendo como fonte um conjunto de ficheiros no formato csv, o maior com aproximadamente 60 milhões de registos. Foi um projecto complexo onde definimos uma metodologia de carregamento deste DW utilizando como ferramenta de integração o MS IS 2005. Tal como descrevi no projecto da Media Capital, trata-se de uma ferramenta complexa para produzir mapeamentos que fujam um pouco ao que consideramos básico. Tentei tirar proveito dos inúmeros objectos que possuí, e alguns, acabaram por ser bastante úteis como as *Slowly Changing Dimensions*, que, eram muito fáceis de manipular. Apesar das inúmeras dificuldades, o resultado final foi do agrado do cliente.

3.3.1 ZON

A minha estadia na ZON consistiu num regresso a área das Telcos. Estive envolvido em dois projectos, um para construção de Dashboards com indicadores de qualidade operacional, tendo efectuado o seu *Roll-Out*, e num outro, ligado a construção de um modelo de análise de Churn onde procedi à sua especificação em conjunto com a área de Marketing interna. Em ambos as tecnologias envolvidas foram o Microstrategy 9i e PL/SQL Oracle.



4. Referências

- [1] <http://www.oracle.com/index.html>
- [2] <http://www.microsoft.com/sqlserver/en/us/default.aspx>
- [3] <http://www.teradata.com/>
- [4] <http://www.sybase.com/>
- [5] <http://www-03.ibm.com/systems/i/software/db2/>
- [6] http://www-03.ibm.com/ibm/history/exhibits/rochester/rochester_4010.html
- [7] <http://www.microstrategy.com>
- [8] <http://ssdi.di.fct.unl.pt/>
- [9] <http://www.microsoft.com/en-us/default.aspx>
- [10] <http://www.novell.com/home/>
- [11] <http://www.novabase.pt/pt/Pages/NovabaseHome.aspx>
- [12] <http://www.staples.pt/>
- [13] <http://www.bimaven.com/>
- [14] Ted Friedman, Mark A. Beyer, Eric Thoo, Magic Quadrant for Data Integration Tools, Gartner, 2009
- [15] http://publib.boulder.ibm.com/infocenter/rentrpt/v1r0m0/index.jsp?topic=%2Fcom.ibm.rational.raer.help.doc%2Ftopics%2Fc_deltaoad.html
- [16] <http://www.sei.cmu.edu/cmmi/>
- [17] <http://www.itil-officialsite.com/>
- [18] Stephen Few, “Information Dashboard Design, The Effective Visual Communication of Data”, O’Reilly,
- [19] <http://www.kimballgroup.com/>
- [20] http://www.kimballgroup.com/html/articles_search/articles1998/9805d05.html
- [21] http://www.kimballgroup.com/html/articles_search/articles2001/0106IE.html?TrkID=IE200106
- [22] <http://ciobriefings.com/Publications/WhitePapers/DesigningtheStarSchemaDatabase/tabid/101/Default.aspx>
- [23] James Richardson, Kurt Schlegel, Rita L. Sallam, Bil Hostmann, Magic Quadrant for Business Intelligence Plataforms, Gartner, 2009
- [24] James Richardson, Rita L. Sallam, Bil Hostmann, Andreas Bitterer, Magic Quadrant for Business Intelligence Plataforms, Gartner, 2010
- [25] <http://www.rittmanmead.com/files/OWB%20Best%20Practices%20%26%20Advanced%20Features.pdf>
- [26] http://download.oracle.com/docs/cd/B13789_01/server.101/b10759/pseudocolumns007.htm
- [27] <http://www.ptsi.pt/>



- [28] <http://www.ptcom.pt/>
- [29] <http://www-01.ibm.com/software/data/infosphere/qualitystage/>
- [30] <http://www.pmo-projects.com/noticiasanteriores/108-o-maior-projecto-de-ti-da-actualidade-em-portugal-esta-a-ser-gerido-atraves-do-metodo-de-earned-value-management-com-o-apoio-da-pmo-consulting.html>
- [31] <http://www.sap.com/portugal/index.epx>
- [32] <http://www-01.ibm.com/software/data/infosphere/datastage/>
- [33] <http://www.esi.pt/>
- [34] <http://www.staples.pt/>
- [35] <http://www.mediacapital.pt/>
- [36] <http://www.acss.min-saude.pt/>
- [37] <http://www.mdn.gov.pt/mdn/pt/>
- [38] <http://www.efacec.pt/>
- [39] <http://www.fnac.pt/>
- [40] <http://www.groupama.pt/>
- [41] <http://www.zon.pt/>
- [42] <http://www.talend.com/products-data-quality/talend-open-profiler.php>



5. Bibliografia adicional

Tendo em consideração o facto desta dissertação, se tratar de um relatório profissional, deixo aqui referencia para alguns livros, que tenho considerado como auxiliares ao desenvolvimento do meu trabalho até a data.

- Kimball, et al. "The Data Warehouse Lifecycle Toolkit", Wiley, 1998
- Kimball, Ross. "The Data Warehouse Toolkit: The Complete Guide to Dimensional Modeling (Second Edition)", Wiley, 2002
- Kimball, Caserta. "The Data Warehouse ETL Toolkit", Wiley. 2004



Anexos

1. ODI

Vou enumerar as mais-valias que encontrei nesta ferramenta relativamente aos concorrentes que fui conhecendo, considerando maioritariamente o conhecimento adquirido *on the job*.

A sua arquitectura por ser E-LT em contraponto com o habitual ETL remove a necessidade de um servidor de ETL *hub*, que tipicamente fica situado entre os servidores de origem e destino. Assim com esta arquitectura, os dados podem ser extraídos das suas fontes e carregados no seu destino, e só posteriormente transformados utilizando o motor de base de dados que acharmos que mais se adequa à transformação requerida, ou que, tenha maior disponibilidade em termos de *Down Time*. Esta solução à cabeça pode apresentar uma mais-valia em termos de redução de custos, não tanto por evitarmos a utilização do servidor de integração para realizarmos as transformações, mas também pelo facto de ao utilizarmos o motor de BD de todos os servidores envolvidos no processo, necessitamos de ter infra-estrutura com menor capacidade, pois mais facilmente fazem um *load balance* desta carga e a tornam numa solução facilmente escalável.

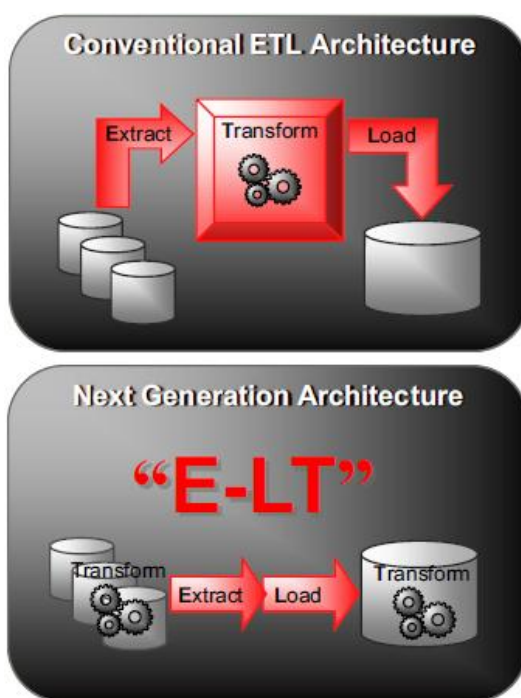


Figura 18 - Arquitectura ETL vs EL-T

A utilização do desenho declarativo ajuda a Abstrair a modelação para um desenho de alto nível, simplificar o número de passos de um processo e gerar automaticamente o fluxo de dados, quaisquer que sejam as fontes ou destinos.

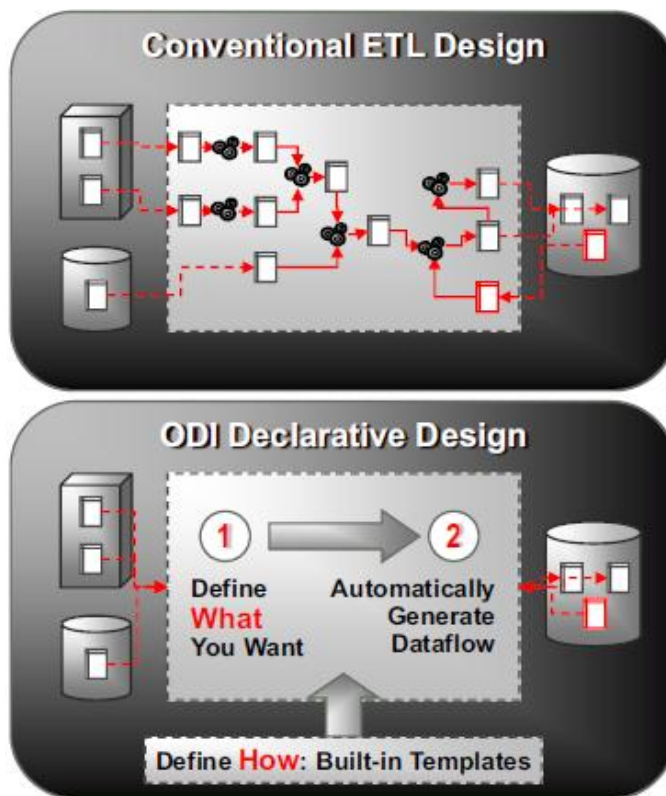


Figura 19 - Design Declarativo

Claro que nem tudo pode ser apontado como bom e perante a minha experiência encontrei uma grande limitação neste desenho declarativo, e que, torna a ferramenta menos eficiente relativamente a outras ofertas que existem no mercado. O facto de não ser possível ter em cada *interface* mais de um destino, ou seja, se quisermos carregar duas estruturas com informação semelhante, temos que criar e executar duas interfaces.

O facto de o ODI ser totalmente *Java based* apresenta como grande vantagem a sua múltipla conectividade, isto porque apenas criando um *driver* utilizando uma *framework* de desenvolvimento e fornecendo a ligação JDBC, é possível ligar a qualquer repositório. Claro que este facto apresenta como grande desvantagem, o facto das mensagens de erro e isto relativamente a muitas outras ferramentas de integração serem pouco intuitivas numa primeira análise, pois validar erros de Java quando se lida com problemas ligados a modelação, não é o maior desejo de um implementador.

Com base neste conceito nasce aquilo que chamamos de agente, que não é mais que aplicação desenvolvida em Java e executável através da *comand line* e que permite fazer o *deploy* de um qualquer processo ETL gerado de forma a poder ser executado remotamente. Mais uma vez será apenas necessário serem configuradas as ligações JDBC destas interfaces remotas, para o local onde se encontram os repositórios que integram ou de onde provém os dados. Isto é muito útil para solucionar problemas de protocolos de segurança em redes de determinadas organizações, porque podemos por vezes não ter acesso a uma determinada fonte ou destino de dados a partir do servidor de integração. Neste caso basta ser fornecida uma outra máquina que possua este acesso, e onde apenas temos de colocar o tal agente, sempre a executar em background. Outra vantagem do conceito de agente é o facto de ser permitida a gestão no ODI de vários agentes e assim fazer o *Load Balance* da carga de um processo entre eles. Para isso, os agentes que executam uma mesma tarefa terão de estar localizados em máquinas diferentes. O processo que faz este controlo consegue ver o número de sessões abertas de cada agente e assim controlar se um agente está



parado ou a executar alguma tarefa pedida. Se a esta funcionalidade adicionarmos o paralelismo nos processos de ETL, permite-nos otimizar os tempos de execução.

Para além do que já descrevi, de referir ainda que o ODI suporta todas as estratégias de carregamento dos dados como o *Bulk Load*, *Changed Data Capture*, *Incremental Update* ou *Slowly Changing Dimension*.

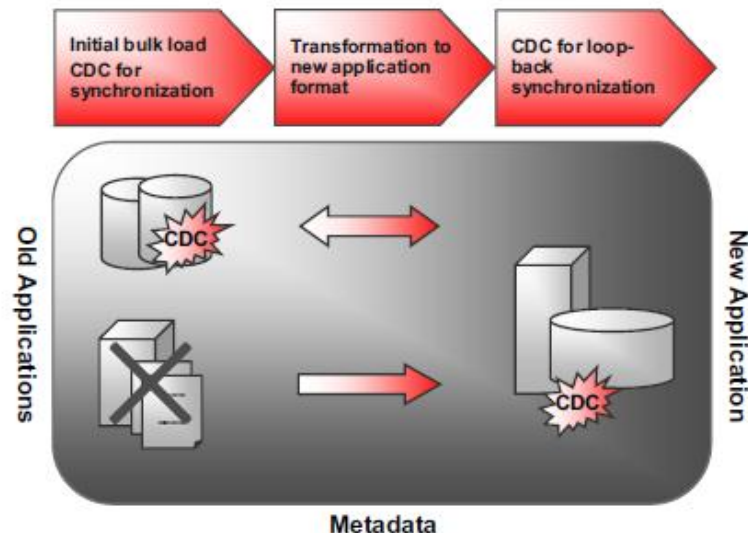
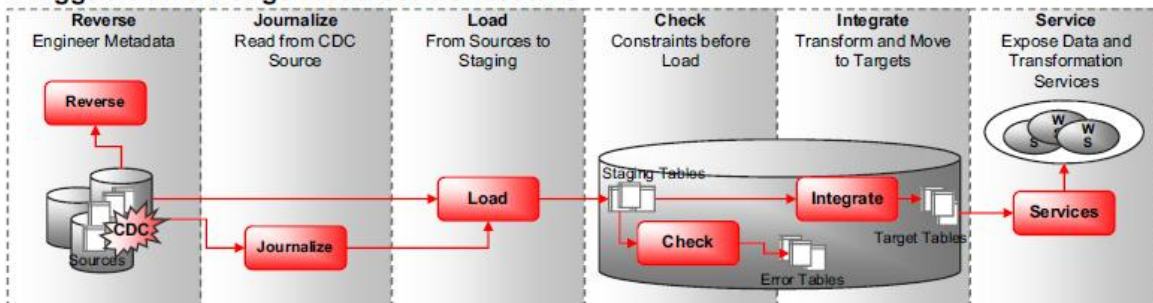


Figura 20 - ODI Operations

Os *Knowledge Modules* (KMs) que sendo *Hot-Pluggable* permitem criar uma arquitectura modular, flexível e extensível que apresenta grandes benefícios pois aproveita ao máximo as optimizações da BD. A construção de processos à medida das *Best Practices* reconhecidas no mercado, facilita a administração dos processos. Estes KMs não são mais que classes criadas numa linguagem da própria ferramenta, e que, contêm uma série de *tasks* base e que podem ser “customizadas” à medida. Aquilo que aconteceu nos projectos que desenvolvemos, foi que antes do início dos desenvolvimentos, foi estipulado tempo para adaptarmos os KMs já existentes às necessidades específicas do projecto, tendo ainda criado alguns KMs de raiz específicos para determinadas tarefas. Esta para mim será a grande mais-valia desta ferramenta de integração, pois não só nos fornece objectos que representam *tasks* parametrizáveis, com muitas concorrentes mas permite “customizar” à medida cada uma no seu mais ínfimo pormenor. Estes KMs têm uma linguagem base utilizada, mas podem invocar código de várias linguagens, não apenas PL/SQL mas Jython, classes Java ou até VB.

Pluggable Knowledge Modules Architecture



Exemplos out-of-the-box de Knowledge Modules



Figura 21 - ODI Hot Pluggable



2. OBIEE

Esta ferramenta tem uma arquitectura cliente-servidor que se divide em 3 camadas, *Physical*, *Business Model and Mapping* e *Presentation*

2.1 Principais funcionalidades

Oracle BI Dashboards

Trata-se de um painel onde é apresentada informação. Um *dashboard* pode ter várias páginas contendo:

- *Reports*
- Gráficos
- Pastas
- Textos
- Conteúdo embebido
- *Links* (navegação assistida, navegação de *Briefing Books*, *Web*)
- Imagens

O conteúdo de um *dashboard* pode ser exportado para formato PDF ou HTML. Desta forma os dados podem ser visualizados sem que o utilizador esteja ligado à rede. É possível guardar uma imagem do *dashboard* num *Briefing Book*, e assim conseguimos guardar o histórico do *Dashboard*.

Briefing Books

Trata-se de uma “*Snapshot*” do *dashboard* ou de um *report*. O seu conteúdo pode ser estático ou dinâmico, actualizado ou agendado, podendo ser entregue através do Oracle BI Delivers.

É possível também, reordenar, apagar ou alterar as propriedades dos conteúdos de um *Briefing Book*, criar *links*, imprimir, descarregar ou ainda adicionar *Briefing Books* a *dashboards*.

Oracle BI Answers

Fornece respostas, permite explorar e interagir com a informação, apresentá-la e visualizá-la por meio de gráficos, tabelas *pivot* e *reports*.

Podemos gravar, organizar e partilhar os resultados. Os resultados poderão ser melhorados através de gráficos, *layout* de resultados, cálculos ou *drill down*. É ainda possível ver quais os *reports* que estão a ser usados num *dashboard*.

Os *reports* e filtros são guardados em pastas que podem ser partilhadas ou simplesmente guardados na área reservada do utilizador.



Para cada *report* é possível criar um *iBot*, que executa o *report* numa determinada altura, e que envia os resultados ou notificação para um ou mais utilizadores.

Oracle Business Intelligence Delivers

É a *interface* utilizada para lançar alertas baseados em resultados analíticos. Podem ser detectados resultados específicos de *reports* e notificar automaticamente utilizadores através da *Web*, *wireless* e telemóvel. O Oracle BI Delivers utiliza *bots* inteligentes (*iBots*) para detectar os resultados.

Os **iBots** são agentes accionados por horários ou eventos, que acedem, filtram ou executam análises nos dados seguindo um critério.

Os utilizadores com permissão podem usar o Oracle BI Delivers com o objectivo de definir as condições para lançar um alerta.

Microsoft Office Plugin

Com este *plugin* é possível importar tabelas e gráficos de *dashboards* para criação de apresentações ou documentos. Depois de importados os dados podem, ver visualizados em modo *offline*.

2.2 Modos de visualização

Existem inúmeras formas de apresentar indicadores em relatórios, através do OBIEE. Nas figuras seguintes apresento o conjunto das mais utilizadas.

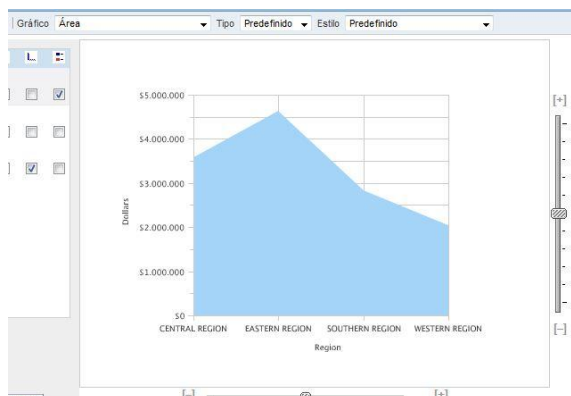


Figura 22 - Gráfico de Área



Figura 25 - Gráfico Linear

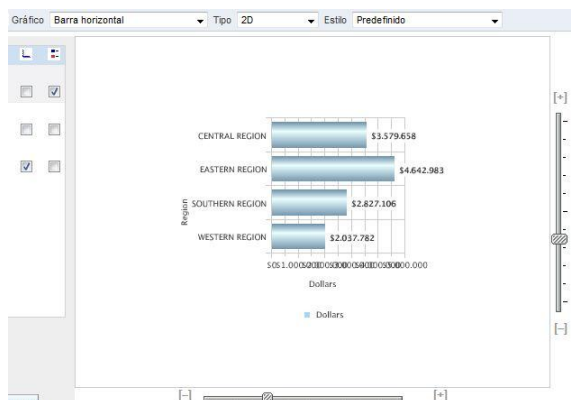


Figura 23 - Gráfico de barras horizontais

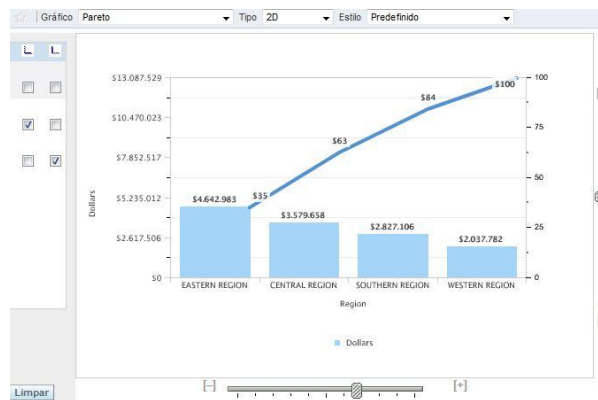


Figura 26 - Gráfico de Pareto

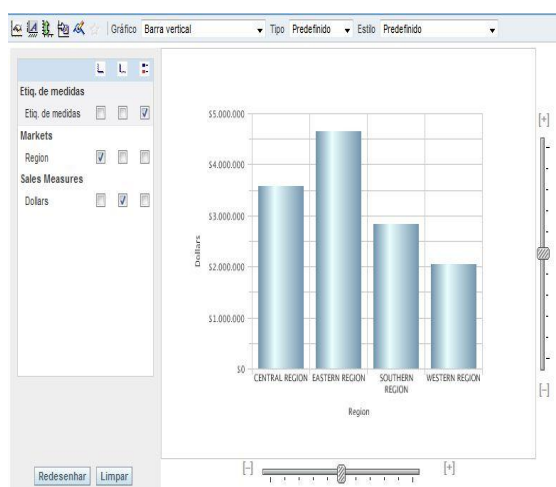


Figura 24 - Gráfico de Barras Verticais

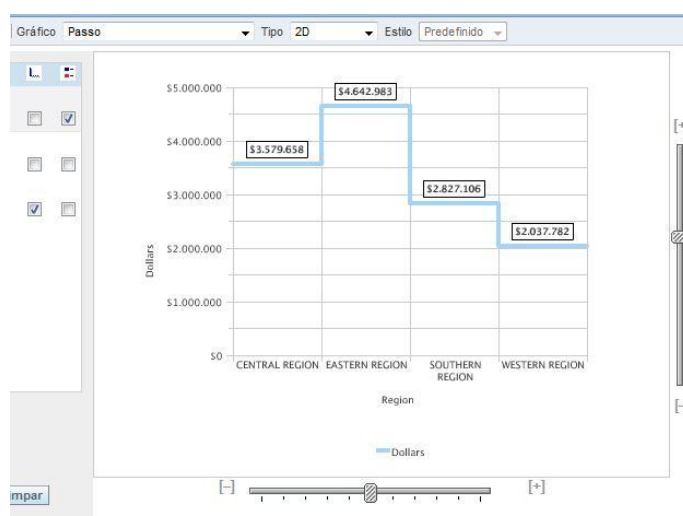


Figura 27 - Gráfico de Passo

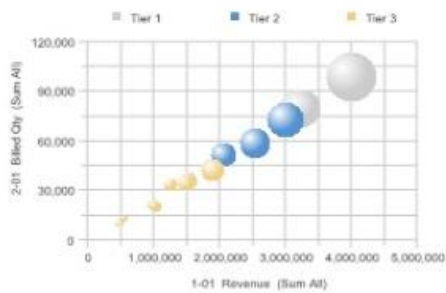


Figura 28 - Gráfico de Bolhas

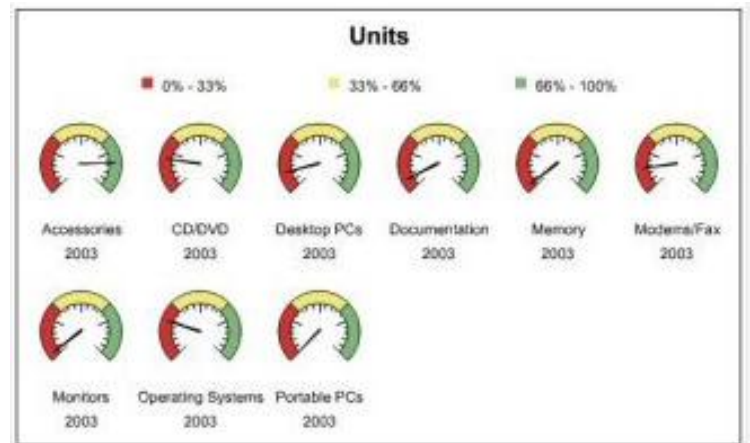


Figura 31 - Gráfico Gauge

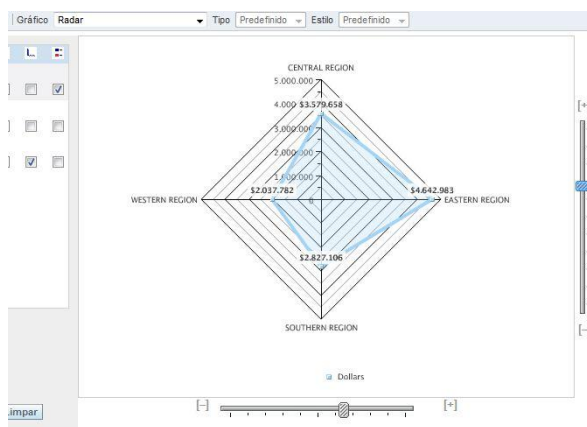


Figura 29 - Gráfico de Radar

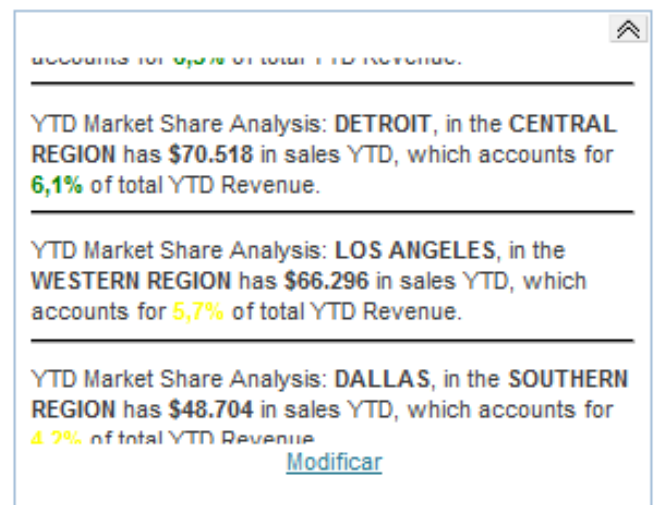


Figura 32 - Gráfico Ticker

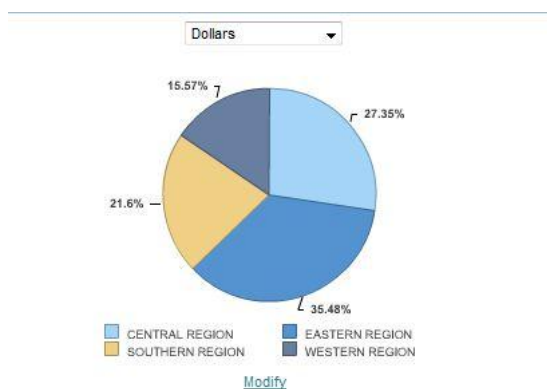


Figura 30 - Gráfico Circular

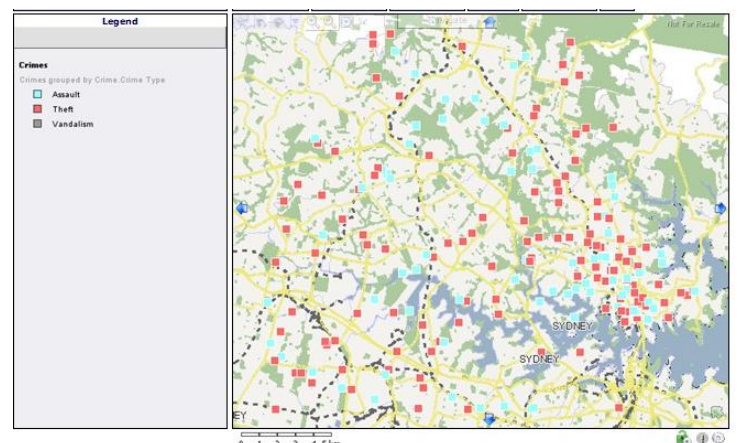


Figura 33 - Map Locations

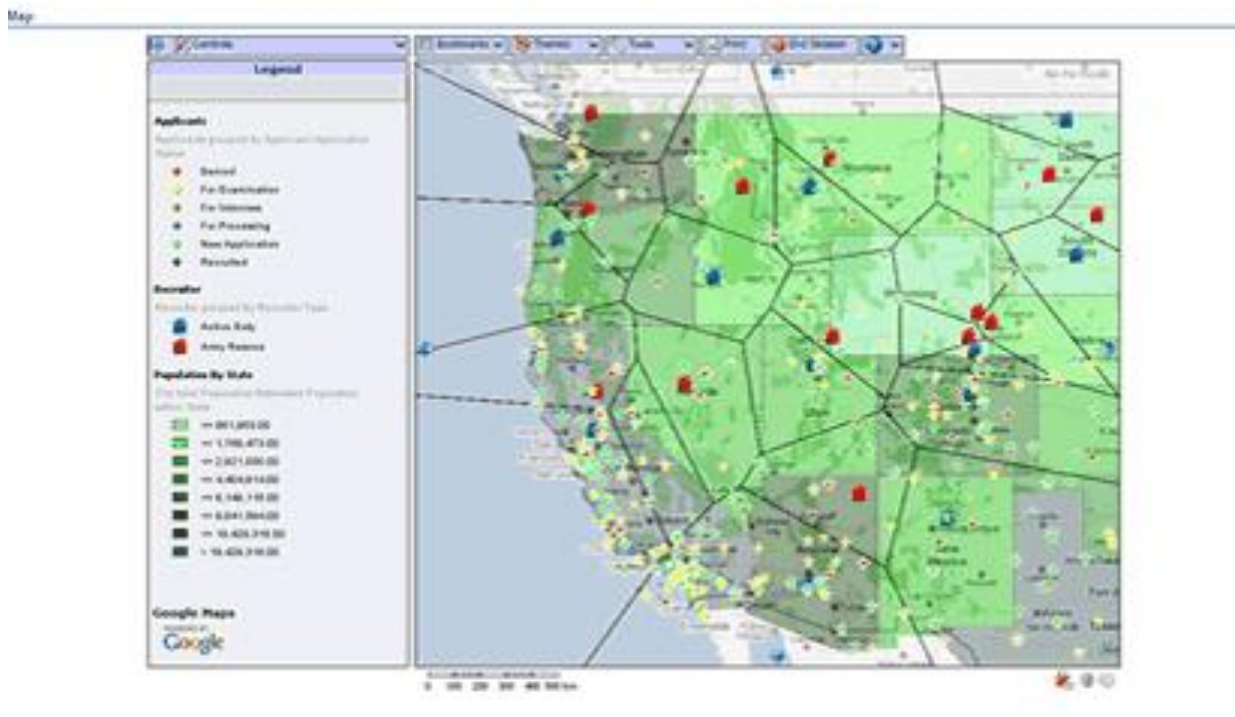


Figura 34 - Map Prompt

2.3 Criação de Indicadores

O processo de criação de indicadores processa-se na camada de *Server* do OBIEE. Nesta temos 3 níveis de *metadata*. Uma que comporta todo o modelo físico, nomeada “*Database*”, outra com o modelo lógico, nomeada “*Business Model*” e uma terceira com o modelo de visualização, que estará disponível na camada de apresentação nomeado “*Presentation Catalog*”.

Através do OBIEE é permitida a geração de um documento XML onde toda a *metadada* é descrita, trata-se de um dicionário de *metadata* que descreve de forma minuciosa todo o Repositório criado.

Assim especificando cada uma das camadas disponíveis e fazendo uma abordagem *bottom-up*. Temos a seguinte estrutura:

- *Database*, onde temos o nome da base de dados fonte, descemos um nível para o *schema* fonte, o modelo lógico, *Logical Framework*. Em cada uma destas estruturas podemos validar os campos que cada uma contém e os respectivos formatos.
- *Business Model*, onde podemos ver separadas:
 - *Dimensions* - Dimensões de análise;
 - *Logical Tables* – Tabelas de Factos e Agregadoras;
 - *Related Objects* – Todas as relações lógicas criadas entre objectos deste modelo lógico.
- *Presentation Catalog*, onde podemos ver as *Presentation Tables*:
 - Se quisermos ver em detalhe qualquer um dos objectos envolvidos conseguimos visualizar a que catalogo pertence cada um, assim como os grupos de utilizadores que têm acesso ao mesmo. Se quisermos validar quem são os utilizadores referenciados



pelo grupo que é visível, basta descermos ao nível deste grupo e podemos ver todos os seus elementos, tal como as respectivas permissões de utilização.

- Na árvore primária, podem adicionar-se linhas de tipo *dummy*, geradas de modo a que na ferramenta possamos ter disponíveis pastas para estruturar a informação.
 - Em qualquer das estruturas disponíveis podemos descer a partir dela até ao nível mais baixo, isto é, conseguimos visualizar qual a fonte física da informação, passando pela lógica igualmente.
 - Nas pastas *dummy* podem existir subpastas contendo indicadores, nestes casos, quando descemos ao nível lógico é possível avaliar toda a construção do indicador, assim como as suas fontes, o nível de segmentação e agregação em que se encontra se este for constituído por vários objectos lógicos, todos os que têm intervenção no cálculo do referido.
- O Dicionário de metada encontra-se totalmente descrito numa extracção em XML em dois formatos possíveis, ou Árvore, *Tree Index* ou Dicionário, *Name Index*.





Dados, Informação, Conhecimento, o Business Intelligence e as suas motivações

André Correia

2011